UNIVERSIDAD DE GUADALAJARA

Red Universitaria e Institución Benemérita de Jalisco

CUCEI

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

División de Tecnologías para la Integración Ciber-Humana

Primer Avance

Alumnos

Casillas Rendón Jennyfher Jacquelinne

Huerta Murillo Alejandro Saul

Lona Avalos Cristian Josué

Programación de Sistemas Avanzados

D01 2026-A

Profesor

López Arce Delgado Jorge Ernesto

Guadalajara, Jalisco, 19 de febrero de 2026

Índice

Índice **2**
Introducción **3**
Objetivo **3**
Objetivo General 3
Objetivos Particulares 3
Desarrollo **4**
Diseño de Topología y Roles 4
Diagrama preliminar de la topología VPN 4
Descripción del rol de cada nodo 4
Implementación de la Infraestructura (WireGuard y Docker) 6
Evidencia de generación de llaves WireGuard 6
Evidencia de conectividad VPN entre al menos dos nodos 7
Evidencia de instalación de Docker y Docker Compose 7
Captura del docker-compose.yml base 10
Descripción de problemas encontrados y decisiones técnicas tomadas 10
Definición y justificación del problema computacional distribuido a implementar 12
Conclusión **16**
Referencias **17**

Introducción

Esta actividad se centra en la implementación de un sistema distribuido dentro de una red privada virtual (VPN), implementando el conjunto de Mandelbrot con Rust y contenedorización con Docker.

Empezaremos hablando sobre la red VPN, la cual es una forma segura al crear un “túnel” entre los dispositivos y el servidor, para esto se crea la encriptación de los datos para que estos viajen de forma segura por la web y además oculta la dirección IP de los dispositivos.

Por otro lado, con Docker podemos empaquetar software en lo que llamamos contenedores, los cuales tienen las dependencias necesarias para ejecutar una aplicación y que funciones de manera estable sin importar la infraestructura.

Lo que se busca es implementar el conjunto de Mandelbrot de forma distribuida, ya que se necesita calcular una fórmula matemática de forma iterada para definir si un número pertenece o no al conjunto y, con este resultado se decide si el pixel de la imagen se pintará de un color o de otro. El conjunto de Mandelbrot crea fractales, es decir, una figura que, al hacer zoom, se crean patrones parecidos al original pero en diferentes escalas.

Por último, hacemos uso de Rust, un lenguaje de programación moderno que se resalta por su seguridad en el manejo de memoria y su eficiencia en Linux, además de que es una buena alternativa para manejar concurrencia sin comprometer la estabilidad del sistema.

Objetivo

Objetivo General

Diseñar, configurar y validar una infraestructura de red privada virtual (VPN) y un entorno de contenedores Docker, que sirva como base para la ejecución distribuida de algoritmos computacionales de alto rendimiento desarrollados en el lenguaje Rust.

Objetivos Particulares

- Infraestructura de Red: Implementar una topología Hub-and-Spoke utilizando el protocolo WireGuard, gestionando manualmente la generación de llaves

criptográficas y el enrutamiento de paquetes entre un nodo central y tres nodos periféricos (peers).

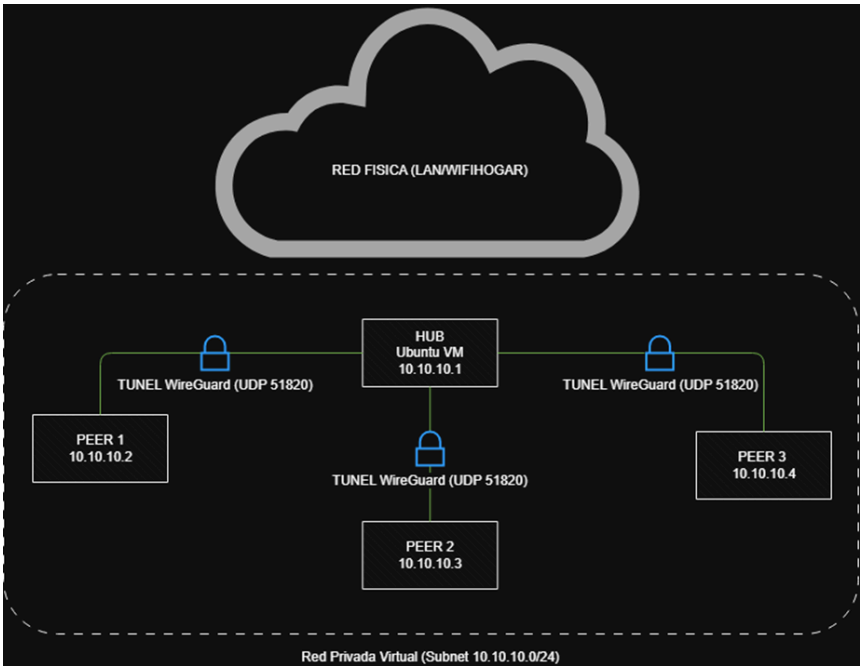
- Contenedorización: Desplegar y orquestar un entorno base de 12 contenedores (4 por cada nodo peer) utilizando Docker Compose, asegurando que la comunicación sea exclusiva a través de las direcciones IP privadas de la VPN (10.10.10.0/24).
- Prototipado Distribuido: Desarrollar el esqueleto funcional (boilerplate) del sistema en Rust, implementando las estructuras de datos necesarias para un modelo de comunicación Coordinador-Worker bajo una arquitectura asíncrona.

Desarrollo

Diseño de Topología y Roles

Se configuró una red privada en el segmento 10.10.10.0/24. El Hub actúa como servidor y los Peers como nodos de cómputo.

Diagrama preliminar de la topología VPN.



Descripción del rol de cada nodo.

Nodo	Nombre del host	IP VPN	Rol
Nodo 1	hub-server	10.10.10.1	Servidor Central: Encargado de enrutar el

			tráfico entre Peers y mantener el tunel WireGuard abierto para conexiones entrantes.
Nodo 2	worker-peer-1	10.10.10.2	Cliente (Spoke): Nodo de cómputo que se conecta al Hub y aloja 4 contenedores worker.
Nodo 3	worker-peer-2	10.10.10.3	Cliente (Spoke): Nodo de cómputo que se conecta al Hub y aloja 4 contenedores worker.
Nodo 4	worker-peer-3	10.10.10.4	Cliente (Spoke): Nodo de cómputo que se conecta al Hub y aloja 4 contenedores worker.

HUB

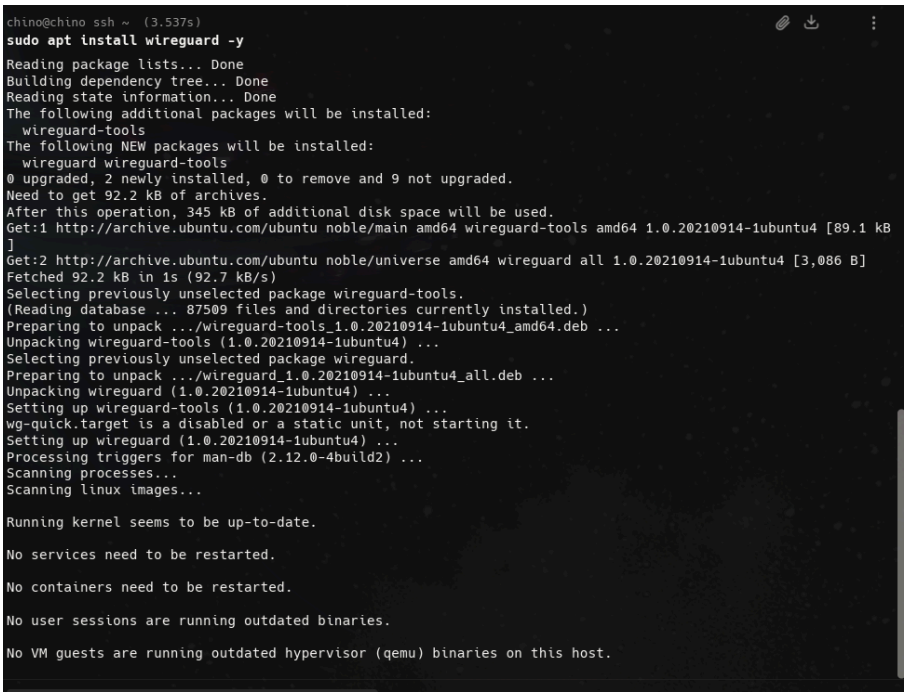
```
sudo wg show
interface: wg0
  public key: GZhvu/mrKc4SX/i5SsBCxVPq4yDcJB1kHGpfTP8DUkk=
  private key: (hidden)
  listening port: 51820

peer: je7ECcbbYFQrEZrEhp4eTG9fXTDoZZxrp6kHNtXJFSE=
  endpoint: 192.168.100.11:54754
  allowed ips: 10.10.10.2/32
  latest handshake: 1 minute, 30 seconds ago
  transfer: 1.27 KiB received, 732 B sent

peer: ikrq7NornZLemQWViCcpRCbtbGHgdakK2H81ybIoawQ=
  allowed ips: 10.10.10.3/32
```

PEERS

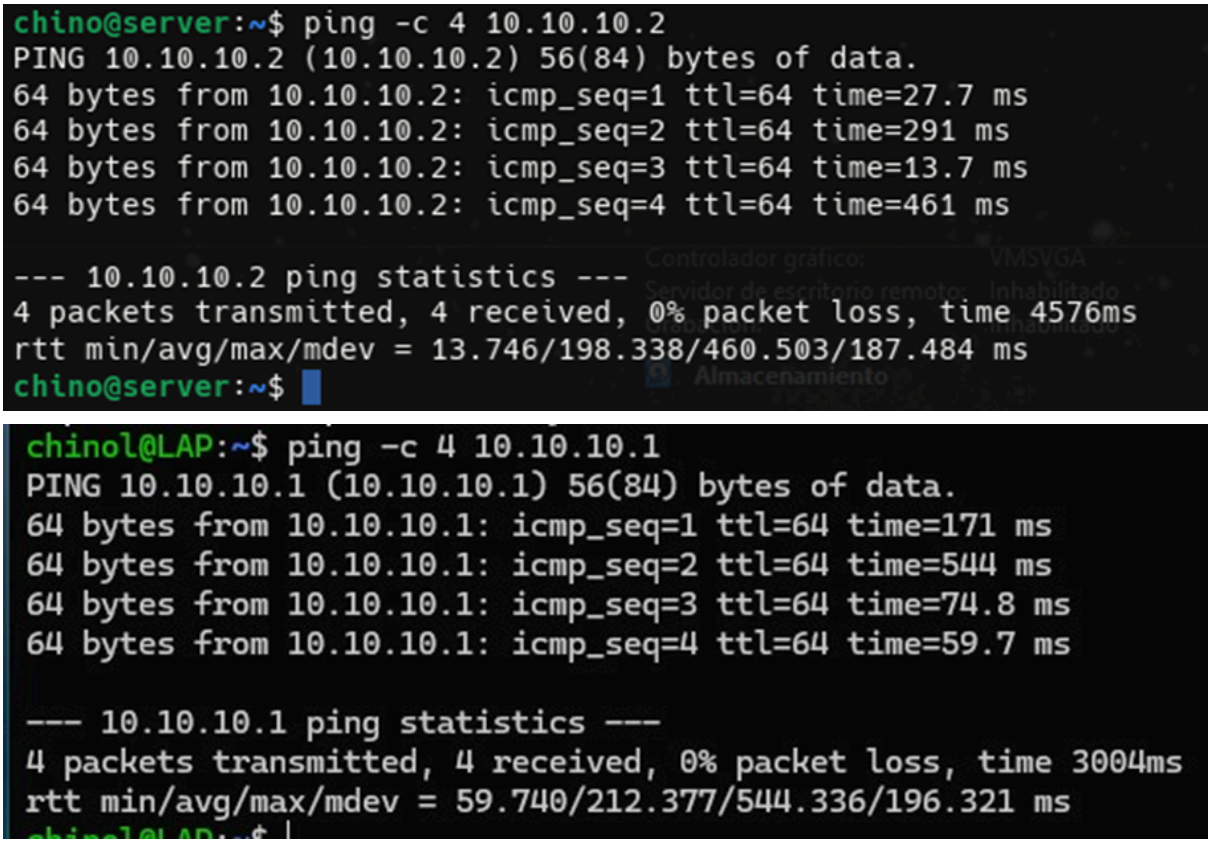
```
interface: wg0
  public key: je7ECcbbYFQrEZrEhp4eTG9fXTDoZZxrp6kHNtXJFSE=
  private key: (hidden)
  listening port: 51982

peer: GZhvu/mrKc4SX/i5SsBCxVPq4yDcJB1kHGpfTP8DUkk=
  endpoint: 192.168.100.138:51820
  allowed ips: 10.10.10.0/24
  latest handshake: 35 minutes, 23 seconds ago
  transfer: 65.34 KiB received, 153.09 KiB sent
  persistent keepalive: every 25 seconds
```

Implementación de la Infraestructura (WireGuard y Docker)

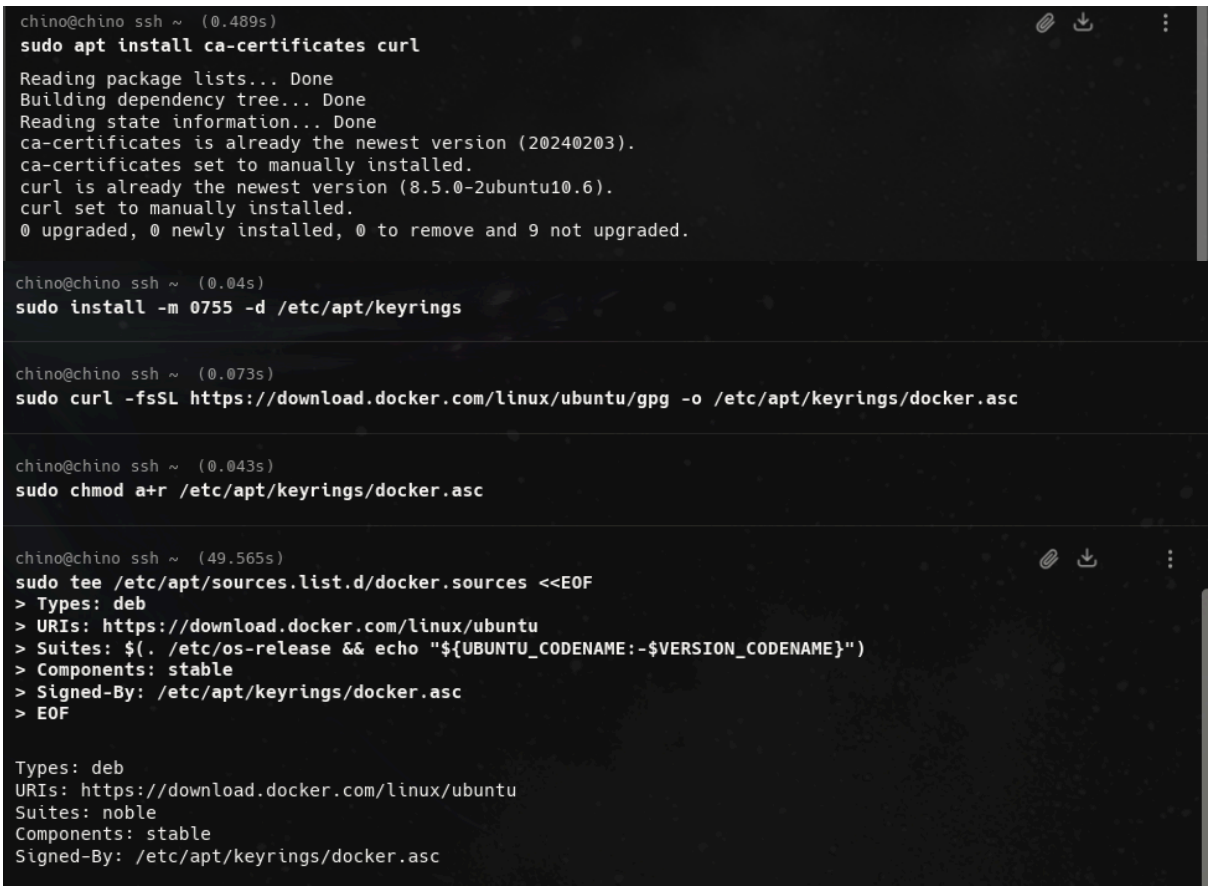
Evidencia de generación de llaves WireGuard.



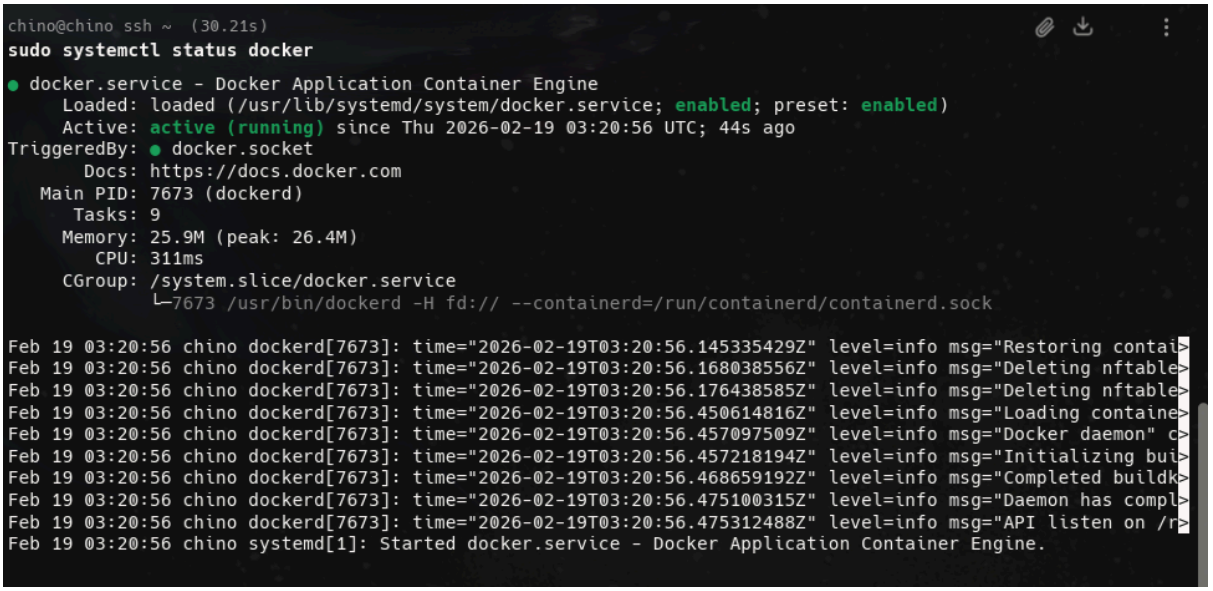
Evidencia de conectividad VPN entre al menos dos nodos.



Evidencia de instalación de Docker y Docker Compose.



```
chino@chino ssh ~  (15.621s)
 sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras libslirp0 pigz slirp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite docker-model-plugin
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin libslirp0 pigz slirp4netns
0 upgraded, 9 newly installed, 0 to remove and 9 not upgraded.
Need to get 96.7 MB of archives.
After this operation, 391 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 2.2.1-1~ubuntu.24.04~noble [23.4
 MB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:3 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:29.2.1-1~ubuntu.24.04~noble [1
6.3 MB]
Get:4 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:29.2.1-1~ubuntu.24.04~noble [22.5
MB]
Get:5 http://archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.31.1-1~ubuntu.24.04~nob
le [20.3 MB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:29.2.1-1~ubuntu.24
.04~noble [6,387 kB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 5.0.2-1~ubuntu.24.04~nob
le [7,721 kB]
Fetched 96.7 MB in 2s (50.2 MB/s)
Selecting previously unselected package containerd.io.
(Reading database ... 87589 files and directories currently installed.)
Preparing to unpack .../0-containerd.io_2.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking containerd.io (2.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../1-docker-ce-cli_5%3a29.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:29.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../2-docker-ce_5%3a29.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:29.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package pigz.
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.31.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.31.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a29.2.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:29.2.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_5.0.2-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-compose-plugin (5.0.2-1~ubuntu.24.04~noble) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../7-libslirp0_4.7.0-1ubuntu3_amd64.deb ...
Unpacking libslirp0:amd64 (4.7.0-1ubuntu3) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../8-slirp4netns_1.2.1-1build2_amd64.deb ...
Unpacking slirp4netns (1.2.1-1build2) ...
Setting up docker-buildx-plugin (0.31.1-1~ubuntu.24.04~noble) ...
Setting up containerd.io (2.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.se
rvice.
Setting up docker-compose-plugin (5.0.2-1~ubuntu.24.04~noble) ...
Setting up docker-ce-cli (5:29.2.1-1~ubuntu.24.04~noble) ...
Setting up libslirp0:amd64 (4.7.0-1ubuntu3) ...
Setting up pigz (2.8-1) ...
Setting up docker-ce-rootless-extras (5:29.2.1-1~ubuntu.24.04~noble) ...
Setting up slirp4netns (1.2.1-1build2) ...
Setting up docker-ce (5:29.2.1-1~ubuntu.24.04~noble) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.7) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```


chino@chino ssh ~ (30.21s)
sudo systemctl status docker
● docker.service - Docker Application Container Engine
Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
Active: active (running) since Thu 2026-02-19 03:20:56 UTC; 44s ago
TriggeredBy: ● docker.socket
Docs: https://docs.docker.com
Main PID: 7673 (dockerd)
Tasks: 9
Memory: 25.9M (peak: 26.4M)
CPU: 311ms
CGroup: /system.slice/docker.service
└─7673 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.145335429Z" level=info msg="Restoring contai
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.168038556Z" level=info msg="Deleting nftable
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.176438585Z" level=info msg="Deleting nftable
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.450614816Z" level=info msg="Loading containe
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.457097509Z" level=info msg="Docker daemon" c
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.457218194Z" level=info msg="Initializing bui
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.468659192Z" level=info msg="Completed buildk
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.475100315Z" level=info msg="Daemon has compl
Feb 19 03:20:56 chino dockerd[7673]: time="2026-02-19T03:20:56.475312488Z" level=info msg="API listen on /r
Feb 19 03:20:56 chino systemd[1]: Started docker.service - Docker Application Container Engine.

Captura del docker-compose.yml base.

```yaml
services:
  cliente_local:
    build:
      context: .
      dockerfile: Dockerfile.client
    network_mode: "host"
    environment:
      - HUB_ADDR=${HUB_ADDR}
      - CLIENT_IP=${CLIENT_IP}
```

```yaml
services:
  hub:
    build:
      context: .
      dockerfile: Dockerfile.hub
    network_mode: "host"

  cliente_local:
    build:
      context: .
      dockerfile: Dockerfile.client
    network_mode: "host"
    environment:
      - HUB_ADDR=10.10.10.1:7878
      - CLIENT_IP=10.10.10.1
```

Descripción de problemas encontrados y decisiones técnicas tomadas.

Desarrollo de Pruebas de Conectividad VPN con WireGuard

Inicié el proyecto realizando pruebas de configuración y conectividad en entornos WSL y máquinas virtuales de Azure. Al intentar implementar el servidor de forma local en el host físico (Windows), enfrenté limitaciones críticas en la gestión de reglas de ruteo y reenvío de puertos (Port Forwarding). Específicamente, el entorno de red de WSL dificultó la redirección de tráfico UDP desde el host hacia la instancia de WSL.

Para mitigar este inconveniente, se optó por desplegar una máquina virtual configurada en modo Bridge. Esta configuración permitió asignar a la VM una dirección IP propia dentro del rango del router físico, facilitando su uso directo como Hub de la VPN sin las restricciones del NAT del host.

Al intentar crear las claves de autenticación, cometí el error de ejecutar el comando de generación por separado, utilizando wg genkey en ambos casos. Esto resultó en la creación de dos llaves privadas distintas (guardando una de ellas erróneamente como si fuera la pública), en lugar de derivar correctamente la llave pública a partir de la privada original. Identifique el problema gracias al comando sudo wg show; al analizar la interfaz, nos dimos cuenta de que la llave configurada en el archivo .conf y el resultado mostrado en la terminal eran completamente distintos. Para solucionarlo, en lugar de generar claves nuevas desde cero, simplemente extraje la llave pública real que arrojaba wg show y actualice con ese valor exacto los registros de la sección [Peer] y el archivo físico de la clave pública.

Una vez establecida la VM en modo Bridge, la conexión seguía sin establecer el handshake. Para verificar qué problemas tenía utilice un comando para revisar la red tcpdump monitoreando el tráfico del puerto UDP 51820. Esto me permitió confirmar que los paquetes del cliente llegaban físicamente al servidor, descartando problemas de ruteo.

Tras revisar minuciosamente los archivos de configuración, descubrí que el error restante se debía a la sintaxis estricta de WireGuard. El archivo wg0.conf del cliente tenía las etiquetas escritas en minúsculas ([interface], Privatekey), lo que provocaba que el servicio ignorara silenciosamente la configuración, causando que el servidor rechazará la conexión. Al corregir las mayúsculas ([Interface], PrivateKey, PublicKey), el túnel se estableció con éxito inmediato, logrando comunicación bidireccional (verificada mediante ping).

Después, creé un archivo docker-compose.yml base que despliega simultáneamente los 4 contenedores worker solicitados. Utilicé una imagen ligera (rust:alpine) con un comando de ejecución continua, dejando los nodos levantados y operativos en la red VPN.

WSL nos causo demasiados problemas tanto que se optó por desechar totalmente el subsistema de nuestra arquitectura y optamos por máquinas virtuales; problemas como incompatibilidad de servicios, overlapping de recursos, inconsistencia en configuraciones etc…

Definición y justificación del problema computacional distribuido a implementar.

El algoritmo pensado es el conjunto de Mandelbrot debido a su alta exigencia computacional al tratar de calcular y renderizar los pixeles que renderizan la imagen del fractal

```
wg0: flags=209<UP,POINTOPOINT,RUNNING,NOARP>  mtu 1420
        inet 10.10.10.1  netmask 255.255.255.0  destination 10.10.10.1
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

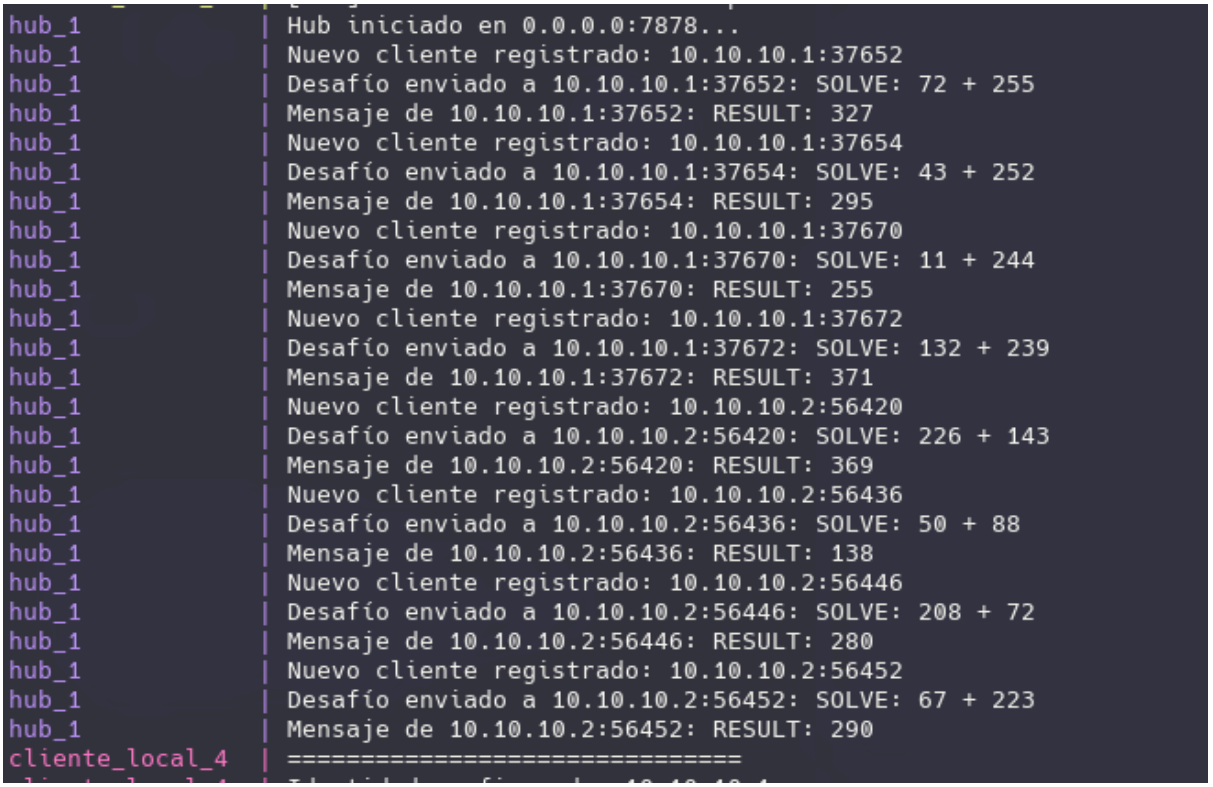
```
xansaul@hub ssh ~/keys  (0.07s)
sudo wg show

interface: wg0
  public key: 0nXts8LurVCWneAVAzO4ndYkKuxVf73JOn7wVR8clA0=
  private key: (hidden)
  listening port: 51820

peer: GQlqG5Az3imx+o3WpkHGP4ihDwtuAmWONU7294B6IyA=
  allowed ips: 10.10.10.2/32

peer: ikrq7NornZLemQWViCcpRCbtbGHgdakK2H81ybIoawQ=
  allowed ips: 10.10.10.3/32
```

```
xansaul@hub ssh ~/VPNDocker/connection-tcp git:(main)  (2.432s)
sudo docker-compose -f docker-compose.hub.yml up -d --scale cliente_local=4

Starting connection-tcp_hub_1 ... done
Starting connection-tcp_cliente_local_1 ... done
Starting connection-tcp_cliente_local_2 ... done
Starting connection-tcp_cliente_local_3 ... done
Creating connection-tcp_cliente_local_4 ... done

xansaul@hub ssh ~/VPNDocker/connection-tcp git:(main)  (0.132s)
docker ps

permission denied while trying to connect to the docker API at unix:///var/run/docker.sock

xansaul@hub ssh ~/VPNDocker/connection-tcp git:(main)  (0.147s)
sudo docker ps

CONTAINER ID   IMAGE                          COMMAND      CREATED          STATUS         PORTS                                       NAMES
9333292a02b4   connection-tcp_cliente_local   "./client"   8 seconds ago    Up 8 seconds                                               connection-tcp_cliente_local_4
c5f59ed0749b   connection-tcp_cliente_local   "./client"   47 minutes ago   Up 9 seconds                                               connection-tcp_cliente_local_3
e5a006f463e5   connection-tcp_cliente_local   "./client"   47 minutes ago   Up 9 seconds                                               connection-tcp_cliente_local_2
51c25b7f6adf   connection-tcp_cliente_local   "./client"   47 minutes ago   Up 9 seconds                                               connection-tcp_cliente_local_1
647581c0217d   connection-tcp_hub             "./hub"      47 minutes ago   Up 9 seconds   0.0.0.0:7878->7878/tcp, [::]:7878->7878/tcp   connection-tcp_hub_1
```

```
xansaul@hub ssh ~/VPNDocker/connection-tcp git:(main)
sudo docker ps
CONTAINER ID   IMAGE                          COMMAND      CREATED         STATUS         PORTS                                       NAMES
30992a7fd6eb   connection-tcp_cliente_local   "./client"   4 seconds ago   Up 4 seconds                                               connection-tcp_cliente_local_4
449c49cdc169   connection-tcp_cliente_local   "./client"   4 seconds ago   Up 4 seconds                                               connection-tcp_cliente_local_1
700e9aa601d4   connection-tcp_cliente_local   "./client"   4 seconds ago   Up 4 seconds                                               connection-tcp_cliente_local_2
5a5b99e80c68   connection-tcp_cliente_local   "./client"   4 seconds ago   Up 4 seconds                                               connection-tcp_cliente_local_3
a102e9f877e7   connection-tcp_hub              "./hub"      5 seconds ago   Up 5 seconds   0.0.0.0:7878->7878/tcp, [::]:7878->7878/tcp   connection-tcp_hub_1

xansaul@hub ssh ~/VPNDocker/connection-tcp git:(main)  (0.638s)
sudo docker-compose -f docker-compose.hub.yml logs
Attaching to connection-tcp_cliente_local_4, connection-tcp_cliente_local_1, connection-tcp_cliente_local_2, connection-tcp_cliente_local_3, connection-tcp_hub_1
cliente_local_3  | Conectando al Hub en: hub:7878...
cliente_local_3  | ¡Conectado exitosamente!
cliente_local_3  | Escribe un mensaje para enviar al Hub (o 'exit' para salir):
cliente_local_4  | Conectando al Hub en: hub:7878...
cliente_local_4  | ¡Conectado exitosamente!
cliente_local_4  | Escribe un mensaje para enviar al Hub (o 'exit' para salir):
cliente_local_1  | Conectando al Hub en: hub:7878...
cliente_local_1  | ¡Conectado exitosamente!
cliente_local_1  | Escribe un mensaje para enviar al Hub (o 'exit' para salir):
cliente_local_2  | Conectando al Hub en: hub:7878...
cliente_local_2  | ¡Conectado exitosamente!
cliente_local_2  | Escribe un mensaje para enviar al Hub (o 'exit' para salir):
hub_1            | Hub iniciado en el puerto 7878...
hub_1            | Nuevo cliente registrado: 172.18.0.3:46184
hub_1            | Nuevo cliente registrado: 172.18.0.4:50992
hub_1            | Nuevo cliente registrado: 172.18.0.5:58592
hub_1            | Nuevo cliente registrado: 172.18.0.6:53376
```

```
xansaul@hub ssh ~/VPNDocker/connection-tcp git:(main)  (0.741s)
sudo docker-compose -f docker-compose.hub.yml logs
Attaching to connection-tcp_cliente_local_4, connection-tcp_cliente_local_3, connection-tcp_cliente_local_2, connection-tcp_cliente_local_1, connection-tcp_hub_1
cliente_local_2  | Identidad configurada: 10.10.10.1
cliente_local_2  | Intentando conectar al Hub (10.10.10.1:7878) vía WireGuard...
cliente_local_2  | ¡Conectado exitosamente!
cliente_local_2  | IP local usada: 10.10.10.1
cliente_local_2  | Puerto local asignado: 39856
cliente_local_2  | Escribe mensajes para enviar (o 'exit' para salir):
cliente_local_4  | Identidad configurada: 10.10.10.1
cliente_local_4  | Intentando conectar al Hub (10.10.10.1:7878) vía WireGuard...
cliente_local_4  | ¡Conectado exitosamente!
cliente_local_4  | IP local usada: 10.10.10.1
cliente_local_4  | Puerto local asignado: 39830
cliente_local_4  | Escribe mensajes para enviar (o 'exit' para salir):
cliente_local_1  | Identidad configurada: 10.10.10.1
cliente_local_1  | Intentando conectar al Hub (10.10.10.1:7878) vía WireGuard...
cliente_local_1  | ¡Conectado exitosamente!
cliente_local_1  | IP local usada: 10.10.10.1
cliente_local_1  | Puerto local asignado: 39816
cliente_local_1  | Escribe mensajes para enviar (o 'exit' para salir):
cliente_local_3  | Identidad configurada: 10.10.10.1
cliente_local_3  | Intentando conectar al Hub (10.10.10.1:7878) vía WireGuard...
cliente_local_3  | ¡Conectado exitosamente!
cliente_local_3  | IP local usada: 10.10.10.1
cliente_local_3  | Puerto local asignado: 39840
cliente_local_3  | Escribe mensajes para enviar (o 'exit' para salir):
hub_1            | Hub iniciado en 0.0.0.0:7878...
hub_1            | Nuevo cliente registrado: 10.10.10.1:39816
hub_1            | Nuevo cliente registrado: 10.10.10.1:39830
hub_1            | Nuevo cliente registrado: 10.10.10.1:39840
hub_1            | Nuevo cliente registrado: 10.10.10.1:39856
hub_1            | Nuevo cliente registrado: 10.10.10.2:56152
hub_1            | Cliente 10.10.10.2:56152 desconectado
hub_1            | Nuevo cliente registrado: 10.10.10.2:41358
hub_1            | Cliente 10.10.10.2:41358 desconectado
hub_1            | Nuevo cliente registrado: 10.10.10.2:33592
hub_1            | Nuevo cliente registrado: 10.10.10.2:33600
hub_1            | Nuevo cliente registrado: 10.10.10.2:33602
hub_1            | Nuevo cliente registrado: 10.10.10.2:33608
```

```
xansaul@spoke1 ssh ~/VPNDocker/connection-tcp git:(main)  (0.383s)
sudo docker compose -f docker-compose.client.yml logs

cliente_local-3  | ==============================
cliente_local-2  | ==============================
cliente_local-2  | Identidad configurada: 10.10.10.2
cliente_local-2  | Intentando conectar al Hub 10.10.10.1:7878 ...
cliente_local-2  | ==============================
cliente_local-2  | Conectado exitosamente al Hub
cliente_local-2  | IP local usada: 10.10.10.2
cliente_local-2  | Puerto local asignado: 33602
cliente_local-2  | ==============================
cliente_local-2  | Escribe mensajes para enviar (exit para salir):
cliente_local-4  | ==============================
cliente_local-4  | Identidad configurada: 10.10.10.2
cliente_local-4  | Intentando conectar al Hub 10.10.10.1:7878 ...
cliente_local-4  | ==============================
cliente_local-4  | Conectado exitosamente al Hub
cliente_local-4  | IP local usada: 10.10.10.2
cliente_local-4  | Puerto local asignado: 33600
cliente_local-4  | ==============================
cliente_local-4  | Escribe mensajes para enviar (exit para salir):
cliente_local-1  | ==============================
cliente_local-1  | Identidad configurada: 10.10.10.2
cliente_local-1  | Intentando conectar al Hub 10.10.10.1:7878 ...
cliente_local-1  | ==============================
cliente_local-1  | Conectado exitosamente al Hub
cliente_local-1  | IP local usada: 10.10.10.2
cliente_local-1  | Puerto local asignado: 33608
cliente_local-1  | ==============================
cliente_local-1  | Escribe mensajes para enviar (exit para salir):
cliente_local-3  | Identidad configurada: 10.10.10.2
cliente_local-3  | Intentando conectar al Hub 10.10.10.1:7878 ...
cliente_local-3  | ==============================
cliente_local-3  | Conectado exitosamente al Hub
cliente_local-3  | IP local usada: 10.10.10.2
cliente_local-3  | Puerto local asignado: 33592
cliente_local-3  | ==============================
cliente_local-3  | Escribe mensajes para enviar (exit para salir):
```


hub_1 | Hub iniciado en 0.0.0.0:7878...
hub_1 | Nuevo cliente registrado: 10.10.10.1:37652
hub_1 | Desafío enviado a 10.10.10.1:37652: SOLVE: 72 + 255
hub_1 | Mensaje de 10.10.10.1:37652: RESULT: 327
hub_1 | Nuevo cliente registrado: 10.10.10.1:37654
hub_1 | Desafío enviado a 10.10.10.1:37654: SOLVE: 43 + 252
hub_1 | Mensaje de 10.10.10.1:37654: RESULT: 295
hub_1 | Nuevo cliente registrado: 10.10.10.1:37670
hub_1 | Desafío enviado a 10.10.10.1:37670: SOLVE: 11 + 244
hub_1 | Mensaje de 10.10.10.1:37670: RESULT: 255
hub_1 | Nuevo cliente registrado: 10.10.10.1:37672
hub_1 | Desafío enviado a 10.10.10.1:37672: SOLVE: 132 + 239
hub_1 | Mensaje de 10.10.10.1:37672: RESULT: 371
hub_1 | Nuevo cliente registrado: 10.10.10.2:56420
hub_1 | Desafío enviado a 10.10.10.2:56420: SOLVE: 226 + 143
hub_1 | Mensaje de 10.10.10.2:56420: RESULT: 369
hub_1 | Nuevo cliente registrado: 10.10.10.2:56436
hub_1 | Desafío enviado a 10.10.10.2:56436: SOLVE: 50 + 88
hub_1 | Mensaje de 10.10.10.2:56436: RESULT: 138
hub_1 | Nuevo cliente registrado: 10.10.10.2:56446
hub_1 | Desafío enviado a 10.10.10.2:56446: SOLVE: 208 + 72
hub_1 | Mensaje de 10.10.10.2:56446: RESULT: 280
hub_1 | Nuevo cliente registrado: 10.10.10.2:56452
hub_1 | Desafío enviado a 10.10.10.2:56452: SOLVE: 67 + 223
hub_1 | Mensaje de 10.10.10.2:56452: RESULT: 290
cliente_local_4 | ==========

```
cliente_local-3  |
cliente_local-3  | [HUB]: Broadcast desde el Hub para 10.10.10.2:56436: RESULT: 138
cliente_local-3  | [HUB]: Broadcast desde el Hub para 10.10.10.2:56436: RESULT: 280
cliente_local-1  | ================================
cliente_local-1  | Identidad configurada: 10.10.10.2
cliente_local-1  | Intentando conectar al Hub 10.10.10.1:7878 ...
cliente_local-1  | ================================
cliente_local-1  | No se pudo conectar: connection timed out. Reintentando en 3s...
cliente_local-1  | No se pudo conectar: connection timed out. Reintentando en 3s...
cliente_local-4  | ================================
cliente_local-4  | Identidad configurada: 10.10.10.2
cliente_local-4  | Intentando conectar al Hub 10.10.10.1:7878 ...
cliente_local-4  | ================================
cliente_local-4  | No se pudo conectar: connection timed out. Reintentando en 3s...
cliente_local-4  | No se pudo conectar: connection timed out. Reintentando en 3s...
cliente_local-4  | Conectado exitosamente al Hub
cliente_local-4  | IP local usada: 10.10.10.2
cliente_local-4  | Puerto local asignado: 56420
cliente_local-4  | ================================
cliente_local-4  | Escribe mensajes para enviar (exit para salir):
cliente_local-4  |
cliente_local-4  | [RETO RECIBIDO]: 226 + 143 = ?
cliente_local-4  | [RETO ENVIADO]: 369
cliente_local-4  |
cliente_local-4  | [HUB]: Broadcast desde el Hub para 10.10.10.2:56420: RESULT: 369
cliente_local-4  | [HUB]: Broadcast desde el Hub para 10.10.10.2:56420: RESULT: 138
cliente_local-4  | [HUB]: Broadcast desde el Hub para 10.10.10.2:56420: RESULT: 280
cliente_local-1  | Conectado exitosamente al Hub
cliente_local-1  | IP local usada: 10.10.10.2
cliente_local-1  | Puerto local asignado: 56452
cliente_local-1  | ================================
cliente_local-1  | Escribe mensajes para enviar (exit para salir):
cliente_local-1  |
cliente_local-1  | [RETO RECIBIDO]: 67 + 223 = ?
cliente_local-1  | [RETO ENVIADO]: 290
cliente_local-1  |
cliente_local-2  | ================================
cliente_local-2  | Identidad configurada: 10.10.10.2
cliente_local-2  | Intentando conectar al Hub 10.10.10.1:7878 ...
cliente_local-2  | ================================
cliente_local-2  | No se pudo conectar: connection timed out. Reintentando en 3s...
cliente_local-2  | No se pudo conectar: connection timed out. Reintentando en 3s...
cliente_local-2  | Conectado exitosamente al Hub
cliente_local-2  | IP local usada: 10.10.10.2
cliente_local-2  | Puerto local asignado: 56446
cliente_local-2  | ================================
cliente_local-2  | Escribe mensajes para enviar (exit para salir):
cliente_local-2  |
cliente_local-2  | [RETO RECIBIDO]: 208 + 72 = ?
cliente_local-2  | [RETO ENVIADO]: 280
cliente_local-2  |
cliente_local-2  | [HUB]: Broadcast desde el Hub para 10.10.10.2:56446: RESULT: 280
```

Conclusión

Ya que la mayoría de los integrantes no teníamos conocimientos tanto de redes como de VPN y Rust, nos centramos más en las pruebas de lo que logramos por partes, lo que nos consumió bastante tiempo. Tuvimos algunos inconvenientes como que no podíamos abrir ciertos puertos en nuestra red del proveedor de internet, dificultando la implementación y

comunicación de los contenedores y la VPN. Otra cuestión que nos dió bastantes problemas fue WSL, el cual reemplazamos por máquinas virtuales con Linux.

Aunque tuvimos muchos tropiezos, de aquí aprendemos a organizarnos e implementar un sistema de trabajo diferente para que el desarrollo no se retrase.

Referencias

Docker Inc. (2024). Docker (Versión 24.0) [Software de computación]. https://www.docker.com

Donenfeld, J. A. (2024). WireGuard [Software de computación]. https://www.wireguard.com

Canonical Ltd. (2024). Ubuntu Server (Versión 22.04 LTS) [Sistema operativo]. https://ubuntu.com/download/server

Wikipedia (2025, 22 diciembre).Benoît Mandelbrot. Wikipedia, La Enciclopedia Libre. https://es.wikipedia.org/wiki/Beno%C3%AEt_Mandelbrot

Microsoft Learn (s. f.). Virtual Private Networking: An Overview. Recuperado de https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/bb742566(v=technet.10)

AWS. (s. f.). ¿Qué es Docker? Recuperado de https://aws.amazon.com/es/docker/

Docker. (s. f.). What is a Container? Recuperado de https://www.docker.com/resources/what-container/