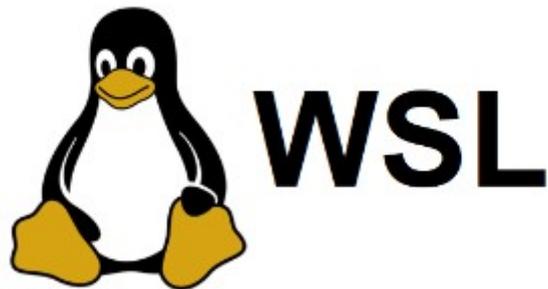


# WSL: Manual no oficial



*Windows Subsystem for Linux*

Carlos García Pinto  
2023

*Este manual es el resultado de un Trabajo de fin de Grado dirigido por Miguel Ángel Ortuño Pérez en la Escuela de Ingeniería de Fuenlabrada de la Universidad Rey Juan Carlos en el año 2023.*

© 2023 Carlos García Pinto. Algunos derechos reservados.

*Este documento se distribuye bajo la licencia Atribución-CompartirIgual 4.0 Internacional de Creative Commons, disponible en <https://creativecommons.org/licenses/by-sa/4.0/deed.es>*

## Contenido

*WSL: Manual no oficial* es una guía diseñada para ayudar a los usuarios que deseen utilizar *Windows Subsystem for Linux*; *WSL* es una característica de Windows que permite ejecutar un entorno Linux dentro de Windows, facilitando la compatibilidad y la ejecución de aplicaciones y comandos de Linux en un entorno Windows. A lo largo de este libro, se proporcionan explicaciones teóricas junto con ejemplos prácticos para demostrar el uso y el funcionamiento adecuado de esta herramienta.

El manual consta de ocho capítulos, cada uno con sus respectivas secciones, que cubren la información más relevante a tener en cuenta sobre el uso y conocimiento de esta aplicación.

**Capítulo 1: Introducción:** En el primer capítulo, se introduce el contexto de Windows Subsystem for Linux (WSL). Este capítulo consta de cuatro secciones que abordan los siguientes temas: ¿Qué es WSL?, una comparativa entre WSL1 y WSL2, y una evaluación de WSL frente a otras alternativas en el ámbito de la virtualización de sistemas operativos.

**Capítulo 2: Instalación de WSL:** El segundo capítulo se centra en explicar de manera secuencial los pasos necesarios para instalar WSL. Aquí, los usuarios pueden aprender sobre la elección de distribuciones de Linux disponibles en WSL y cómo ejecutar una distribución predeterminada en su sistema.

**Capítulo 3: Interoperabilidad Windows-Linux:** En el tercer capítulo, se profundiza en la interoperabilidad entre Windows y Linux que ofrece WSL. Se exploran aspectos como la gestión de usuarios en WSL, cómo compartir archivos y directorios entre ambos sistemas, cómo manejar permisos en directorios y archivos de Windows y Linux, y la posibilidad de deshabilitar la interoperabilidad.

**Capítulo 4: Servicios:** El cuarto capítulo se enfoca en las utilidades y servicios que WSL proporciona para mejorar la experiencia de los usuarios al utilizar Windows junto con una distribución de Linux. Este capítulo aborda temas como el uso de SSH y sus derivados, así como la virtualización a través de contenedores.

**Capítulo 5: Archivos de configuración de WSL:** El quinto capítulo está dedicado a explicar los principales archivos de configuración de WSL: `wsl.conf` y `.wslconfig`. Estos archivos permiten personalizar y ajustar diversos aspectos del entorno de WSL, ya sea de manera global para todas las distribuciones o específicamente para cada una.

**Capítulo 6: Conexión de dispositivos USB:** El sexto capítulo aborda la cuestión de la conexión de dispositivos USB a WSL. Aunque esta funcionalidad no está disponible de forma nativa, se mencionan soluciones como el proyecto de código abierto `usbipd-win` y la posibilidad de acceder desde Windows a discos externos formateados en `ext4`.

**Capítulo 7: Ejecución de aplicaciones de GUI para Linux:** El séptimo capítulo explica cómo WSL permite ejecutar aplicaciones de interfaz gráfica de usuario (GUI) de Linux en Windows. Se detallan los requisitos necesarios para lograrlo y se explican aspectos como el inicio de aplicaciones de Linux desde el menú Inicio de Windows, la

gestión de aplicaciones en la barra de tareas y la capacidad de copiar y pegar entre aplicaciones de Windows y Linux.

**Capítulo 8: Comparativa de rendimiento entre WSL y Linux nativo:** En el octavo y último capítulo del manual, se realizan comparativas de rendimiento entre WSL y Linux nativo en áreas como el rendimiento de la CPU, las interfaces de red, las estructuras de almacenamiento y las operaciones de lectura y escritura. Estas comparativas ayudarán a los usuarios a evaluar las diferencias entre ambos entornos y a tomar decisiones informadas. El capítulo concluye con un resumen de los resultados de las pruebas.

# Índice

1. Introducción.....	1
1.1. ¿Qué es WSL?.....	1
1.2. WSL1 vs WSL2.....	1
1.3. WSL frente a otras alternativas.....	2
2. Instalación de WSL.....	4
2.1. Prerrequisitos.....	4
2.2. Instalación.....	4
2.3. Comandos básicos.....	7
2.4. Distribuciones de Linux en WSL.....	7
2.4.1. Instalación de varias distribuciones Linux.....	7
2.4.2. Importación y exportación de distribuciones.....	9
2.4.3. Ejecutar una distribución predeterminada.....	9
2.4.4. Eliminar distribución.....	10
3. Interoperabilidad Windows-Linux.....	11
3.1. Usuarios en WSL.....	11
3.1.1. Restablecer contraseña.....	12
3.2. Compartir archivos y directorios.....	12
3.2.1. Acceder al sistema de ficheros de Linux (WSL) desde Windows.....	12
3.2.2. Acceder al sistema de ficheros de Windows desde Linux (WSL).....	13
3.3. Permisos de directorios y ficheros de Windows y Linux.....	14
3.3.1. Acceso a archivos creados en Windows desde WSL.....	15
3.3.2. Acceso a archivos creados en WSL desde Windows.....	16
3.4. Deshabilitación de la interoperabilidad.....	17
4. Servicios.....	18
4.1. SSH y derivados.....	18
4.1.1. Conexión SSH desde máquina externa de la misma red a WSL.....	19
4.2. Virtualización: contenedores.....	25
4.2.1. Docker.....	25
4.2.2. Podman.....	26
5. Archivos de configuración de WSL.....	27
5.1. wsl.conf.....	27
5.1.1. Secciones configurables disponibles.....	28
5.2. .wslconfig.....	31
5.2.1. Secciones configurables disponibles.....	31
6. Conexión de dispositivos USB.....	34
6.1. usbipd-win.....	34
6.1.1. Requisitos previos.....	34
6.1.2. ¿Qué es USB/IP?.....	34
6.1.3. Configuración.....	35
6.1.4. Comprobación.....	35
6.2. Acceso desde Windows a un disco externo formateado en ext4.....	36
7. Ejecución de aplicaciones de GUI para Linux.....	38
7.1. Lanzar aplicaciones gráficas de WSL.....	38
7.2. Lanzar archivos de vídeo o audio desde WSL.....	40
8. Comparativa de rendimiento entre WSL y Linux Nativo.....	41
8.1. Benchmarks de CPU.....	41
8.2. Comparación de interfaces de red y Benchmarks de red.....	49

8.3. Comparación de estructuras de almacenamiento y Benchmarks de operaciones de lectura y escritura.....55

8.4. Conclusión de los Benchmarks.....59

## Índice de figuras

Figura 2.2.1: Comando de instalación de WSL.....	5
Figura 2.2.2: Creación de usuario Linux.....	5
Figura 2.2.3: Instalación WSL finalizada.....	6
Figura 2.2.4: WSL lanzado desde PowerShell.....	6
Figura 2.4.1: Distribuciones disponibles en WSL.....	7
Figura 2.4.2: Varias distribuciones de Linux en la misma máquina.....	8
Figura 2.4.3: Consultar distribuciones instaladas.....	8
Figura 2.4.4: Lanzar distribución desde explorador de Windows.....	9
Figura 3.2.1: Acceso a ficheros de WSL desde Windows.....	13
Figura 3.2.2: Acceso a ficheros de Windows desde WSL.....	13
Figura 3.3.1: Habilitar metadatos de WSL.....	14
Figura 3.3.2: Añadir metadatos a un archivo de WSL.....	15
Figura 3.3.3: Visualizar metadatos de un archivo en WSL.....	15
Figura 3.3.4: Cambio de permisos a ficheros creados en Windows desde WSL.....	16
Figura 3.3.5: Acceso a fichero creado sin permisos en WSL desde Windows.....	17
Figura 4.1.1: Conexión SSH a máquina externa.....	18
Figura 4.1.2: Montaje de directorio remoto vía sshfs.....	19
Figura 4.1.3: Comando ssh-keygen.....	19
Figura 4.1.4: Comando scp.....	19
Figura 4.1.5: Redirección de puerto de SSH a WSL.....	20
Figura 4.1.6: Configuración de regla de Firewall I.....	21
Figura 4.1.7: Configuración de regla de Firewall II.....	22
Figura 4.1.8: Configuración de regla de Firewall III.....	22
Figura 4.1.9: Configuración de regla de Firewall IV.....	23
Figura 4.1.10: Configuración de regla de Firewall V.....	23
Figura 4.1.11: Averiguar IP de la conexión de red.....	24
Figura 4.1.12: Conexión SSH establecida desde máquina externa.....	24
Figura 4.2.1: Prueba de tipo "hola, mundo" en Docker.....	25
Figura 4.2.2: Prueba contenedor Docker.....	26
Figura 4.2.3: Prueba contenedor Podman.....	26
Figura 5.1.1: Contenido del fichero wsl.conf.....	27
Figura 5.1.2: Ejemplo de configuración del fichero wsl.conf.....	30
Figura 5.2.1: Ejemplo de configuración del fichero .wslconfig.....	33
Figura 6.1.1: Listar y asociar dispositivo USB a WSL.....	35
Figura 6.1.2: Dispositivo USB conectado a WSL.....	36
Figura 6.1.3: Dispositivo USB desconectado de WSL.....	36
Figura 6.2.1: Listado de unidades de disco conectadas a Windows.....	36
Figura 6.2.2: Mensaje de error al montar una unidad USB.....	37
Figura 7.1.1: Servidor gráfico de WSL: Wayland.....	38
Figura 7.1.2: Lanzar aplicación gráfica desde WSL.....	39
Figura 7.1.3: Lanzar aplicación gráfica de WSL desde Windows.....	39
Figura 7.2.1: Redirección de audio y vídeo de WSL a Windows.....	40
Figura 8.1.1: Benchmark de CPU I en WSL.....	42
Figura 8.1.2: Benchmark de CPU I en Linux nativo.....	43
Figura 8.1.3: Gráfica de Benchmark de CPU I (1).....	44
Figura 8.1.4: Gráfica de Benchmark de CPU I (2).....	44
Figura 8.1.5: Benchmark de CPU II en WSL.....	45
Figura 8.1.6: Benchmark de CPU II en Linux nativo.....	45
Figura 8.1.7: Gráfica de Benchmark de CPU II.....	46

Figura 8.1.8: Benchmark de CPU III en WSL.....	47
Figura 8.1.9: Benchmark de CPU III en Linux nativo.....	47
Figura 8.1.10: Gráfica de Benchmark de CPU III.....	48
Figura 8.2.1: Ping en WSL.....	49
Figura 8.2.2: Ping en Linux nativo.....	49
Figura 8.2.3: Interfaces de red visibles en WSL.....	49
Figura 8.2.4: Interfaces de red visibles en Linux nativo.....	50
Figura 8.2.5: Conexiones TCP y UDP visibles en WSL.....	51
Figura 8.2.6: Conexiones TCP y UDP visibles en Linux nativo.....	51
Figura 8.2.7: Benchmark de red I: Servidor iperf en WSL (1).....	53
Figura 8.2.8: Benchmark de red I: Cliente iperf en Linux nativo.....	53
Figura 8.2.9: Benchmark de red I: Servidor iperf en WSL (2).....	53
Figura 8.2.10: Benchmark de red I: Cliente iperf en WSL.....	53
Figura 8.2.11: Gráfica de Benchmark de red I: Cliente iperf en Linux nativo.....	54
Figura 8.2.12: Gráfica de Benchmark de red I: Cliente iperf en WSL.....	54
Figura 8.3.1: Comparación de dispositivos de bloque I en WSL.....	55
Figura 8.3.2: Comparación de dispositivos de bloque I en Linux nativo.....	56
Figura 8.3.3: Benchmark de tiempo de escritura en WSL.....	56
Figura 8.3.4: Benchmark de tiempo de escritura en Linux nativo.....	57
Figura 8.3.5: Gráfica de Benchmark de tiempo de escritura.....	57
Figura 8.3.6: Benchmark de tiempo de lectura en WSL.....	58
Figura 8.3.7: Benchmark de tiempo de lectura en Linux nativo.....	58
Figura 8.3.8: Gráfica de Benchmark de tiempo de lectura.....	59

# 1. Introducción

## 1.1. ¿Qué es WSL?

WSL (Windows Subsystem for Linux) es una herramienta de Windows 10 y posteriores que permite a los usuarios ejecutar un entorno de Linux, pudiendo ejecutar la mayoría de herramientas de línea de comandos y prácticamente cualquier aplicación en modo texto y algunas aplicaciones gráficas, utilidades y aplicaciones directamente en Windows y sin la sobrecarga de una máquina virtual tradicional o una configuración de arranque dual (tener dos sistemas operativos en la máquina mediante una partición de disco). En la primera versión de WSL, esto se conseguía mediante la implementación de una capa de compatibilidad entre ambos sistemas operativos.

En el año 2020 se lanzó la segunda versión de WSL, WSL2, que contaba con diferentes mejoras con respecto a la versión inicial. En esta segunda versión, WSL ya cuenta con un verdadero núcleo de Linux ejecutado por un hipervisor como las máquinas virtuales tradicionales que permite ejecutar una instancia completa de Linux junto con Windows. Esto significa que se puede acceder a todas las funciones y servicios de Linux, incluidos los controladores del *kernel*, y ejecutar aplicaciones de Linux en el entorno de Windows. La experiencia de uso es muy similar a la de un Linux nativo. Por omisión la distribución empleada es Ubuntu, aunque también soporta diferentes distribuciones como Debian o kali-linux, entre otras (ver sección 2.1).

Aunque es una tecnología similar a la de máquinas virtuales tradicionales, al estar integrada por Microsoft dentro de Windows, es mucho más eficiente y fácil de usar. En lo sucesivo, el término WSL referenciará a la última versión a no ser que se especifique lo contrario.

## 1.2. WSL1 vs WSL2

En el año 2020 fue lanzado WSL2, una versión mejorada de WSL que ofrece varias mejoras con respecto al rendimiento y a la funcionalidad mostrando una experiencia casi idéntica al uso de un Linux nativo ya que esta última versión incorpora un *kernel* de Linux completo.

Son varias las diferencias que hay entre ambas versiones pero las más importantes a destacar son las siguientes:

- **Arquitectura subyacente.** WSL se basa en una capa de compatibilidad que se ejecuta en el núcleo de Windows, mientras que WSL2 utiliza una máquina virtual Linux completa para ejecutar las distribuciones de Linux. Debido a esta diferencia, WSL2 proporciona un mejor rendimiento y una mayor compatibilidad con las aplicaciones de Linux que requieren un *kernel* de Linux completo.
- **Kernel de Linux.** WSL utiliza el *kernel* de Windows para ejecutar aplicaciones de Linux, mientras que WSL2 utiliza un *kernel* de Linux real. Esto permite que WSL2

proporcione una mayor compatibilidad con aplicaciones de Linux que dependen de características específicas del *kernel* de Linux.

- **Gestión de recursos.** WSL2 utiliza una máquina virtual Linux completa, lo que permite una mejor gestión de recursos, como la memoria y el uso de la CPU. Además, WSL2 permite el uso de varios procesadores y se puede configurar para utilizar más o menos recursos según sea necesario.
- **Redirección de puertos.** WSL2 proporciona una mejor redirección de puertos, lo que facilita la configuración de aplicaciones de servidor de Linux que se ejecutan en la máquina virtual.

A modo de conclusión, en WSL1 Microsoft reimplementó todo el sistema como una capa de compatibilidad. Sin embargo, en WSL2, hay un verdadero núcleo de Linux virtualizado (esto es, corriendo sobre una máquina virtual). Pero solo se virtualiza el núcleo. El resto de componentes del sistema, como el disco o la red, se implementan como una capa de compatibilidad.

### 1.3. WSL frente a otras alternativas

Si lo que se busca es poder trabajar con una máquina Linux, la primera alternativa preferible a considerar es instalar Linux nativo en una partición de disco. Cuando por alguna razón no se quiere hacer, existen múltiples opciones para lograr este objetivo. Por ejemplo, es posible conseguirlo mediante el uso de máquinas virtuales tradicionales (VirtualBox o VMWare) o acceder a una máquina Linux desde una máquina Windows de manera remota (por ejemplo, vía VNCWeb o ssh) entre otras posibilidades.

En el caso particular de acceso vía VNCWeb, son varios los inconvenientes que podrían surgir durante su uso. VNCWeb necesita una conexión a internet permanente y de buena calidad. Si la velocidad proporcionada no fuera lo suficientemente rápida, la experiencia de uso no sería la más adecuada ya que podría presentar ciertos inconvenientes como, por ejemplo, experimentar mucho retraso con el teclado, ratón y el refresco de la imagen. Además, VNCWeb no permite por sí solo intercambiar ficheros con la máquina local.

WSL es una alternativa bastante reciente (versión 2 de WSL: año 2020) que se suma a este abanico de posibilidades y posiblemente sea la más cercana a un Linux nativo tradicional.

Si bien es cierto que la alternativa a escoger dependerá del usuario y sus necesidades, WSL ofrece una serie de ventajas a tener en cuenta a la hora de elegir entre una opción u otra.

- **Núcleo de Linux completo.** WSL2 utiliza una máquina virtual que cuenta con un *kernel* de Linux completo.

- **Interoperabilidad Windows-Linux.** WSL está integrado en Windows y ofrece la posibilidad de trabajar con las utilidades de una máquina Linux dentro de una máquina Windows.
- Con WSL es posible trabajar de manera simultánea con ambos sistemas operativos, es decir, permite la posibilidad de usar las aplicaciones y de acceder a los archivos de Windows desde Linux (y viceversa). Esto también es posible con una máquina virtual tradicional como VirtualBox, la diferencia es la comodidad: en WSL el usuario no tiene que hacer apenas ninguna configuración adicional.
- WSL ofrece un entorno independiente de Windows. En los capítulos sucesivos se analizará con más detalle qué características pueden compartir y cuáles no.
- **Más ligero.** WSL utiliza una tecnología de virtualización ligera llamada *Virtual Machine Platform* (Plataforma de Máquina Virtual) que se basa en *Hyper-V*, el hipervisor de virtualización de Microsoft. Esta tecnología de virtualización ligera permite ejecutar un *kernel* de Linux dentro de Windows sin la necesidad de iniciar una máquina virtual completa. En lugar de eso, se crea un entorno aislado que proporciona una capa de compatibilidad entre el sistema operativo host (Windows) y el entorno Linux.
- **Tiempo de arranque.** Al ser más ligero, el tiempo que necesita para arrancar es menor que el necesitado por otras alternativas.
- **Recursos necesarios.** WSL necesita menos recursos (CPU, memoria y almacenamiento) que una máquina virtual completa y que la instalación de un sistema operativo en nativo.
- **Fácil introducción a Linux para usuarios de Windows.** No hay necesidad de instalar un nuevo sistema operativo para poder usar Linux en una máquina Windows; es mucho más rápido. Además, no es necesario crear particiones en el disco, lo cual es una tarea potencialmente peligrosa.

## 2. Instalación de WSL

### 2.1. Prerrequisitos

1. **Versión de Windows.** Para poder instalar WSL en una máquina Windows, es necesario contar con una versión de Windows 10 o posterior.
2. **Espacio suficiente.** Como cualquier otra aplicación que se instala en una máquina, es necesario tener el suficiente espacio en disco para su instalación. El tamaño inicial en disco suele ser relativamente pequeño, generalmente alrededor de 200-300 MB. Sin embargo, a medida que se vaya utilizando y personalizando el entorno de WSL, es posible que el tamaño en disco aumente debido a la instalación de paquetes adicionales, programas y datos. También es importante tener en cuenta que el tamaño que ocupa en disco WSL puede variar según la distribución y los paquetes que se instalen. Algunas distribuciones, como Ubuntu, pueden requerir más espacio en disco debido a la inclusión de paquetes y bibliotecas adicionales.

### 2.2. Instalación

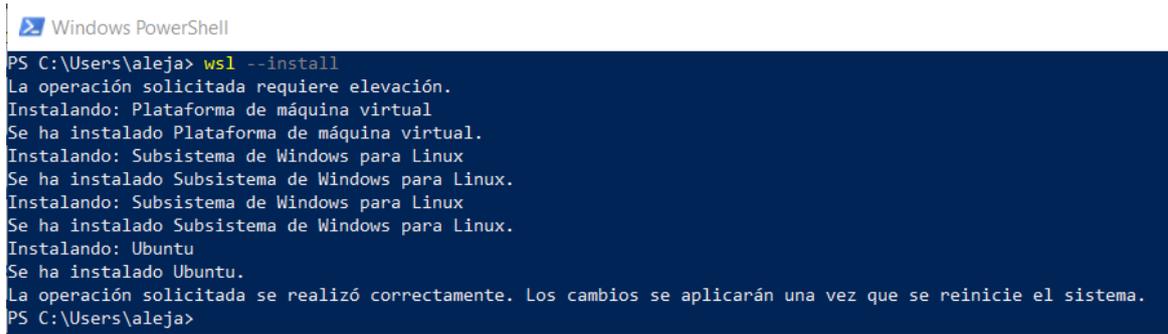
El proceso de instalación de WSL puede realizarse de manera rápida y sencilla mediante los intérpretes de comandos de Windows a través de PowerShell o del símbolo de sistema de Windows ejecutados como administrador.

#### 1) Iniciar PowerShell o símbolo de sistema de Windows.

Para ello, dirigirse al explorador de Windows y escribir en la barra de búsqueda *PowerShell* o *cmd*. A continuación, hacer clic derecho sobre el programa en cuestión y dentro del menú de opciones que se despliega, clicar sobre *ejecutar como administrador*.

**2) Instalar WSL.** Para instalar WSL, ejecutar el siguiente comando de instalación en la PowerShell o cmd de Windows: `wsl -install`

- Este comando habilitará las características necesarias para ejecutar WSL e instalará la distribución Ubuntu de Linux.
- Ubuntu es la distribución por defecto que se instalará; sin embargo, este comportamiento puede cambiarse si se desea instalar otra distribución.



```
Windows PowerShell
PS C:\Users\aleja> wsl --install
La operación solicitada requiere elevación.
Instalando: Plataforma de máquina virtual
Se ha instalado Plataforma de máquina virtual.
Instalando: Subsistema de Windows para Linux
Se ha instalado Subsistema de Windows para Linux.
Instalando: Subsistema de Windows para Linux
Se ha instalado Subsistema de Windows para Linux.
Instalando: Ubuntu
Se ha instalado Ubuntu.
La operación solicitada se realizó correctamente. Los cambios se aplicarán una vez que se reinicie el sistema.
PS C:\Users\aleja>
```

*Figura 2.2.1: Comando de instalación de WSL*

**3) Reiniciar el equipo.** Después de que haya terminado de ejecutarse el anterior comando, es necesario reiniciar el equipo para que se apliquen los cambios. Una vez arrancado de nuevo, se lanzará de manera automática el terminal de WSL. A continuación, el terminal pedirá que se introduzca un nombre de usuario UNIX y, posteriormente, una contraseña para dicho usuario para la distribución de Linux recién instalada; este usuario es propio de WSL, es decir, no tendrá ningún tipo de relación con los usuarios que previamente hubiera en Windows.



```
Ubuntu
Ubuntu ya está instalado.
Iniciando Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username:
```

*Figura 2.2.2: Creación de usuario Linux*

Además, este usuario será considerado el administrador de Linux y tendrá privilegios especiales. Una vez terminado todo lo explicado, se mostrará la siguiente información:

```
carlos@LAPTOP-BEA9J9V1: ~
Ubuntu ya está instalado.
Iniciando Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: carlos
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

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

This message is shown once a day. To disable it please create the
/home/carlos/.hushlogin file.
carlos@LAPTOP-BEA9J9V1:~$
```

*Figura 2.2.3: Instalación WSL finalizada*

Llegados a este punto, ya es posible trabajar con WSL. También es posible iniciar WSL desde el intérprete de comandos de Windows con el comando `wsl ~`

```
carlos@DESKTOP-EE08Q76: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\carli> wsl ~
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

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

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/carlos/.hushlogin file.
carlos@DESKTOP-EE08Q76: $
```

*Figura 2.2.4: WSL lanzado desde PowerShell*

El directorio en que el que se iniciará por omisión es `/home/<usuario>`.

Para consultar dudas adicionales sobre el proceso de instalación o comandos útiles de WSL (a ejecutar desde el intérprete de comandos de Windows), consultar el siguiente enlace: información adicional.

## 2.3. Comandos básicos

Todos los comandos que comienzan por la instrucción `wsl`, son comandos a ejecutar desde el intérprete de comandos de Windows.

- Actualizar WSL a la versión más reciente: `wsl --update`
- Comprobación del estado de WSL: `wsl --status`
- Comprobación de la versión de WSL: `wsl --version`
- Ayuda de WSL: `wsl --help`
- Finalizar la ejecución de todas las distribuciones: `wsl --shutdown`
- Finalizar la ejecución de una distribución específica:  
`wsl --terminate <Distribution Name>`

## 2.4. Distribuciones de Linux en WSL

WSL permite trabajar con múltiples distribuciones de Linux de manera simultánea. Aunque con el comando de instalación `wsl --install` la distribución de Linux que se instala por defecto es Ubuntu, es posible cambiar este comportamiento indicando al comando de instalación en un segundo parámetro cuál es la que se quiere instalar; el comando resultante es  
`wsl --install <Distribution Name>`

### 2.4.1. Instalación de varias distribuciones Linux

Una vez instalado WSL, es posible añadir nuevas distribuciones al entorno. Para consultar las distribuciones que hay disponibles, se puede ejecutar el comando `wsl --list --online`

Actualmente, estas son las distribuciones posibles a instalar mediante el comando `wsl --install <Distribution Name>`



```
Windows PowerShell
PS C:\Users\carli> wsl --list --online
A continuación, se muestra una lista de las distribuciones válidas que se pueden instalar.
Instalar con "wsl.exe --install <Distro>".

NAME                                FRIENDLY NAME
-----                                -
Ubuntu                               Ubuntu
Debian                               Debian GNU/Linux
kali-linux                           Kali Linux Rolling
Ubuntu-18.04                          Ubuntu 18.04 LTS
Ubuntu-20.04                          Ubuntu 20.04 LTS
Ubuntu-22.04                          Ubuntu 22.04 LTS
OracleLinux_7_9                       Oracle Linux 7.9
OracleLinux_8_7                       Oracle Linux 8.7
OracleLinux_9_1                       Oracle Linux 9.1
SUSE-Linux-Enterprise-Server-15-SP4   SUSE Linux Enterprise Server 15 SP4
openSUSE-Leap-15.4                   openSUSE Leap 15.4
openSUSE-Tumbleweed                  openSUSE Tumbleweed
```

Figura 2.4.1: Distribuciones disponibles en WSL

Cuando se instalan diferentes distribuciones de Linux en WSL, cada una de ellas se considera una instancia independiente. Esto significa que cada distribución de Linux en WSL tiene su propio sistema de archivos raíz, su propio entorno de ejecución y sus propias configuraciones.

La independencia entre las distribuciones de Linux en WSL implica que se puede trabajar con ellas de forma aislada, como si fueran sistemas operativos separados. Cada distribución tiene su propio conjunto de paquetes instalados, su propia configuración de red, su propio conjunto de usuarios y permisos, entre otros aspectos. Esto permite que se pueda tener diferentes distribuciones de Linux en WSL para diversos propósitos. Además, las distribuciones de Linux en WSL pueden ejecutarse de forma simultánea, lo que significa que se pueden abrir varias ventanas de terminal, cada una ejecutando una distribución diferente pudiendo alternar entre ellas y realizar tareas específicas en cada una sin interferir en el funcionamiento de las demás.

Sin embargo, también es importante tener en cuenta que aunque las distribuciones de Linux en WSL son independientes entre sí, comparten el mismo *kernel* subyacente de Windows. Esto significa que comparten recursos y dependen del *kernel* de Windows para acceder al hardware y realizar ciertas operaciones del sistema.

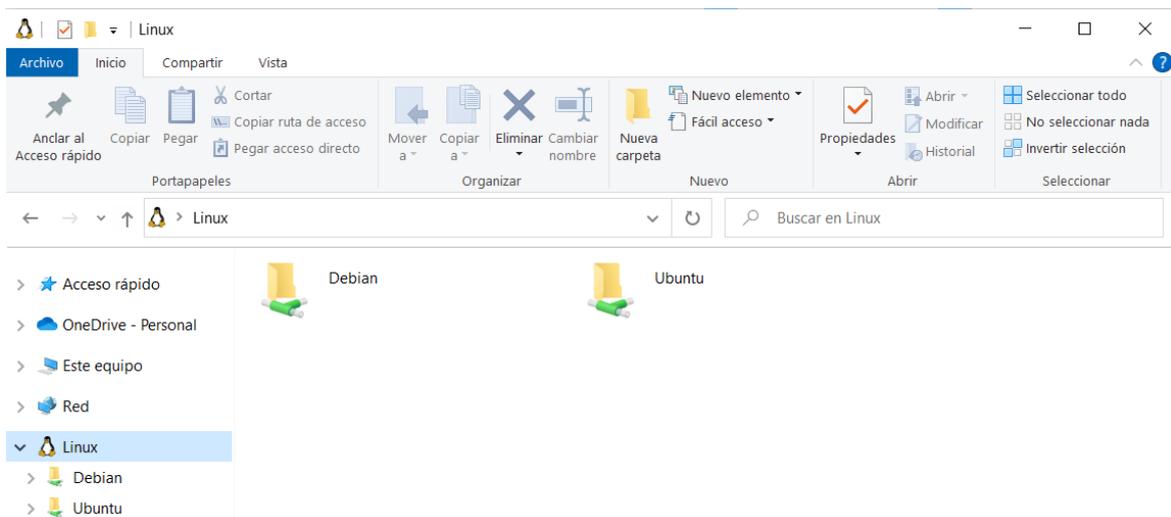


Figura 2.4.2: Varias distribuciones de Linux en la misma máquina

El comando `wsl --list --verbose` mostrará las distribuciones de Linux instaladas en la máquina.

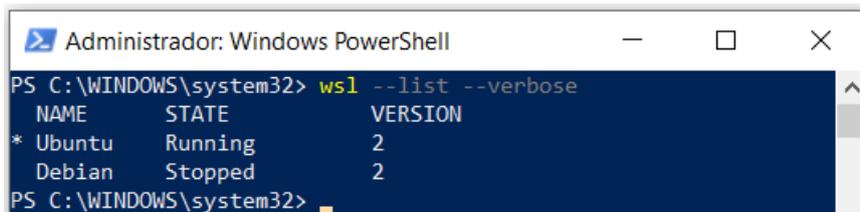


Figura 2.4.3: Consultar distribuciones instaladas

## 2.4.2. Importación y exportación de distribuciones

La importación y exportación de una distribución en WSL se refiere a la capacidad de crear un archivo comprimido que contiene todos los archivos y configuraciones de una distribución instalada en WSL, y luego poder importar ese archivo en otro sistema para instalar y utilizar la misma distribución.

Comando de exportación: `wsl --export <Distribution Name> <FileName>`

Comando de importación:

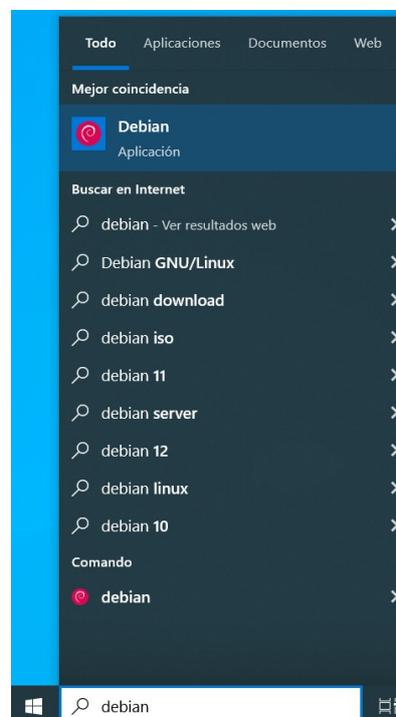
`wsl --import <Distribution Name> <Install Location> <FileName>`

## 2.4.3. Ejecutar una distribución predeterminada

Si se desea trabajar con WSL desde su propio terminal, la distribución que iniciará por defecto será la primera que haya sido instalada. Para cambiar este comportamiento, se puede ejecutar desde el intérprete de comandos de Windows el comando `wsl --set-default <Distribution Name>` para elegir cuál es la distribución que se quiere utilizar por defecto al lanzar WSL.

Por otra parte, si se quiere trabajar desde el intérprete de comandos de Windows con WSL lanzando una distribución concreta, utilizar el comando `wsl --distribution <nombre_distribución>`

También es posible lanzar una distribución concreta desde el explorador de Windows escribiendo sobre la barra de búsqueda el nombre de la distribución que se quiera ejecutar; esta distribución será lanzada con el usuario que tenga establecido por defecto.



*Figura 2.4.4: Lanzar distribución desde explorador de Windows*

Para cambiar el usuario predeterminado para una distribución, ejecutar el siguiente comando:

```
<Distribution Name> config --default-user <Username>
```

Otra alternativa posible es lanzar desde el intérprete de comandos de Windows una distribución de Linux específica con un usuario específico mediante el comando

```
wsl --distribution <Distribution Name> --user <Username>
```

#### **2.4.4. Eliminar distribución**

Para desinstalar una distribución de WSL, hay que tener en cuenta que una vez que se haya desinstalado, todos los datos, la configuración y el software asociados a esa distribución se perderán de manera permanente.

Si de todas formas se quiere seguir adelante con el proceso de desinstalación, ejecutar el comando

```
wsl --unregister <Distribution Name>
```

Tras haber ejecutado el anterior comando, el comando `wsl --list --verbose` no mostrará la distribución que haya sido eliminada.

### 3. Interoperabilidad Windows-Linux

WSL permite la posibilidad de trabajar con Windows y Linux de manera simultánea.

No obstante, es importante tener en cuenta que WSL es una herramienta que se ejecuta dentro de Windows y que hereda componentes del sistema operativo. Aunque existen áreas de Windows que pueden ser gestionadas desde WSL, hay otras que no pueden ser modificadas directamente desde este entorno.

Una de las ventajas de WSL es la posibilidad de acceder al sistema de archivos de Windows desde la distribución de Linux en WSL. Esto significa que es posible acceder, leer, escribir y manipular archivos y directorios tanto de Windows como de la distribución de Linux en WSL, lo que facilita compartir y trabajar con datos entre ambos sistemas operativos.

Por otro lado, las configuraciones de Windows, como la configuración del sistema, las preferencias del panel de control, los ajustes de seguridad y las configuraciones del cortafuegos, generalmente no se ven afectadas directamente por las acciones realizadas en WSL. Para modificar estas configuraciones, es necesario realizar los cambios directamente en el entorno de Windows.

Por todo ello, es importante tener en cuenta que WSL no reemplaza por completo a un entorno de Linux nativo. Aunque ofrece una gran interoperabilidad, existen algunas limitaciones y diferencias sutiles en el comportamiento y la configuración en comparación con una instalación de Linux nativa. Sin embargo, WSL brinda una solución sólida para aprovechar las ventajas de ambos sistemas operativos y trabajar de manera eficiente en un entorno integrado.

#### 3.1. Usuarios en WSL

Para entender el funcionamiento de los usuarios en WSL, es necesario tener en cuenta lo siguiente:

- **Los usuarios de WSL y Windows son totalmente independientes.** Los usuarios de WSL y los usuarios de Windows son distintos y no están directamente vinculados. Cada distribución de Linux en WSL tiene su propio conjunto de usuarios y contraseñas independientes. Cualquier administración de usuarios, creación de cuentas o cambios en las contraseñas deben realizarse dentro del entorno de Linux en WSL y no afectarán a los usuarios del sistema operativo Windows.  
Por ello, cuando se instala una distribución Linux en WSL, se pide crear un usuario que será el primer y único usuario que habrá por el momento hasta que se añadan nuevos dado que no hay ninguno previamente ya que los usuarios de Windows no son visibles en WSL.
- **Cada distribución de Linux que se ejecuta en WSL tiene sus propias cuentas de usuario y contraseñas de Linux.** Será necesario configurar una cuenta de usuario de Linux cada vez que se reinstale, se restablezca o se agregue una distribución.

- **Usuario root y comando sudo.** En WSL, ser usuario root o ejecutar un comando con permisos de root mediante el comando `sudo`, es equivalente a su uso en una máquina Linux. Es decir, sirve para ejecutar ciertos comandos que no se podrían ejecutar sin tener los permisos suficientes como puede ser el caso del comando `sudo apt upgrade`.  
Todos los cambios que se realicen siendo usuario root solo afectan a la distribución Linux dentro de WSL por lo que el usuario root en WSL no tiene relación con el usuario *Administrador* en el sistema operativo Windows. Si se quiere hacer algún cambio en Windows, ha de hacerse directamente sobre Windows.

### 3.1.1. Restablecer contraseña

En caso de haber olvidado la contraseña de un usuario, el procedimiento para recuperarla es idéntico al realizado en Linux teniendo en cuenta que, para recuperarla, es necesario iniciar WSL siendo usuario root.

Los pasos a realizar son los siguientes:

- **Lanzar WSL siendo root.** Para ello, ejecutar WSL desde el intérprete de comandos de Windows con el comando `wsl -u root`

El resto de pasos a realizar a partir de aquí son los mismos que habría que seguir si se estuviera trabajando en un Linux.

- **Actualizar contraseña.** Ejecutar el comando `passwd <username>` donde `username` es el nombre del usuario que ha perdido la contraseña.  
A continuación, será necesario escribir una nueva contraseña y posteriormente confirmarla.

## 3.2. Compartir archivos y directorios

WSL facilita el intercambio de archivos y directorios entre Windows y Linux. Es posible acceder a los archivos de Windows desde la distribución de Linux en WSL y viceversa. Esto permite trabajar con archivos en ambos sistemas operativos sin problemas y utilizar herramientas específicas de cada uno en los archivos compartidos.

Además, WSL se integra con el subsistema de Windows, lo que significa que es posible acceder a recursos de Windows, como dispositivos de hardware, controladores y servicios, desde la distribución de Linux en WSL. Esto proporciona una experiencia más completa y permite utilizar aplicaciones y servicios de Windows junto con las capacidades de Linux.

### 3.2.1. Acceder al sistema de ficheros de Linux (WSL) desde Windows

Es posible consultar ficheros desde Windows que han sido creados en WSL desde el explorador de archivos de Windows de manera rápida y sencilla. Para ello, solo hay que abrir el explorador de archivos de Windows y dirigirse dentro del panel de navegación (barra lateral izquierda) al apartado *Linux*. Una vez dentro, hay que buscar el directorio en el que se encuentre el fichero que se quiere consultar.

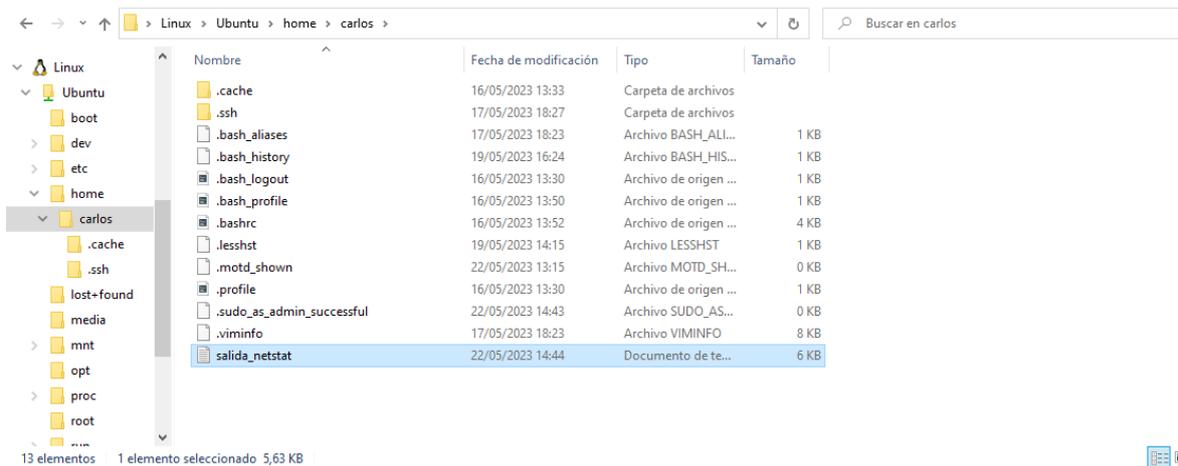


Figura 3.2.1: Acceso a ficheros de WSL desde Windows

Además, ejecutando en WSL el comando `explorer.exe .` (incluyendo el punto final), se abrirá de manera automática una pestaña del explorador de archivos de Windows en la ruta donde se estaba trabajando en WSL para visualizarlo desde Windows.

### 3.2.2. Acceder al sistema de ficheros de Windows desde Linux (WSL)

El acceso a los ficheros dependerá de dónde se haya instalado WSL. Por defecto, WSL se instala en la unidad del sistema de Windows (generalmente, C:). Para acceder al sistema de ficheros de Windows desde WSL, hay que acceder al disco de Windows, (normalmente C:), que se encuentra en `/mnt/c` siendo `c` el disco donde está instalado el sistema de ficheros de WSL.

Cabe destacar que Windows es insensible a mayúsculas y minúsculas por lo que da igual referirse al disco de Windows con `c:` o `C:` aunque normalmente se escribe en mayúsculas. Sin embargo, Unix/Linux sí que es sensible al uso de mayúsculas y minúsculas por lo que, para acceder al disco de Windows, hay que escribir el nombre de la unidad en minúsculas.

Por ejemplo, si se quisiera acceder a los archivos que se encuentran en el directorio escritorio de la máquina Windows desde WSL, habría que acceder a la ruta `/mnt/c/Users/<nombre_usuario_Windows>/<Directorio>` siendo `nombre_de_usuario_Windows` el nombre de usuario de Windows y `Directorio` el directorio escritorio. El comando resultante es `cd /mnt/c/Users/<nombre_usuario_Windows>/Desktop`

```
carlos@LAPTOP-BEA9J9V1: /mnt/c/Users/aleja/Desktop
carlos@LAPTOP-BEA9J9V1:~$ cd /mnt/c/Users/aleja/Desktop/
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$ ls
carpeta_escritorio  desktop.ini
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$
```

Figura 3.2.2: Acceso a ficheros de Windows desde WSL

### 3.3. Permisos de directorios y ficheros de Windows y Linux

El sistema de permisos en WSL sobre los ficheros y directorios siguen el mismo mecanismo que en el caso de uso de un Linux nativo. Sin embargo, a la hora de acceder a los archivos de Windows desde WSL, los permisos de archivo se calculan a partir de los permisos de Windows o se leen de los metadatos que WSL ha agregado al archivo, aunque estos metadatos no están habilitados de forma predeterminada.

Cuando los metadatos están habilitados como opción de montaje en WSL, se pueden agregar e interpretar atributos extendidos en archivos de Windows para proporcionar los permisos del sistema de archivos de Linux. WSL puede agregar cuatro atributos extendidos de NTFS:

- **\$LXUID** → Id. del propietario del usuario.
- **\$LXGID** → Id. del propietario del grupo.
- **\$LXMOD** → Modo de archivo (octales y tipo de permisos de sistemas de archivos; por ejemplo: 0777).
- **\$LXDEV** → Dispositivo, si es un archivo de dispositivo.

Para habilitar los metadatos en WSL, es necesario agregar la opción *metadata* en el archivo `/etc/wsl.conf` y reiniciar WSL para que los cambios surtan efecto (ver capítulo 5). Dicho fichero debería contener una línea como esta:

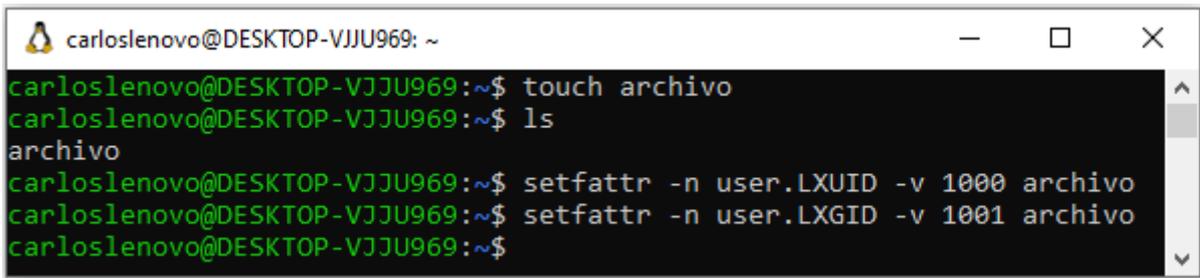
A screenshot of a terminal window. The title bar shows the user 'carloslenovo@DESKTOP-VJJU969' and the current directory '~'. The terminal content shows the configuration for the 'automount' section: 'options = "metadata,uid=1000,gid=1000,umask=022"'. The terminal background is black with white text.

```
carloslenovo@DESKTOP-VJJU969: ~  
[automount]  
options = "metadata,uid=1000,gid=1000,umask=022"
```

Figura 3.3.1: Habilitar metadatos de WSL

La anterior línea hace lo siguiente: cuando se monten unidades de Windows en WSL, se habilitarán los metadatos para que los permisos de archivo de Linux se apliquen correctamente a los archivos de Windows. Además, el propietario del usuario y el propietario del grupo predeterminados para los archivos y directorios montados se establecerán en 1000, y la máscara de creación de archivos se establecerá en 022.

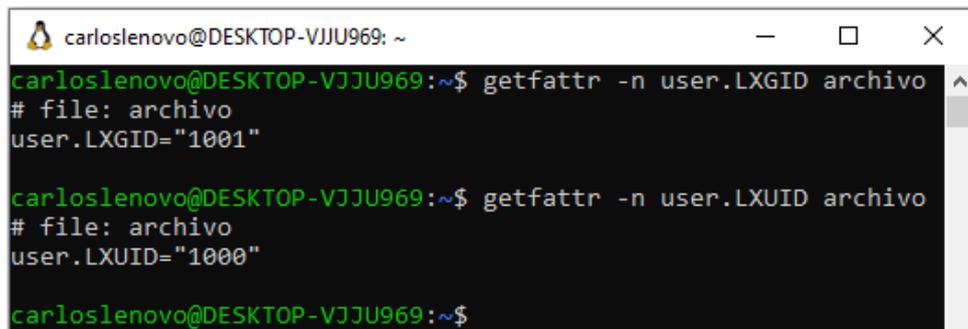
Una vez habilitados los metadatos, se puede usar el comando `setfaattr` (instalar paquete *attr*) para establecer los valores de los atributos extendidos. Por ejemplo, para establecer el propietario del usuario y el propietario del grupo de un archivo a 1000 y 1001 respectivamente:



```
carloslenovo@DESKTOP-VJJU969: ~  
carloslenovo@DESKTOP-VJJU969:~$ touch archivo  
carloslenovo@DESKTOP-VJJU969:~$ ls  
archivo  
carloslenovo@DESKTOP-VJJU969:~$ setfattr -n user.LXUID -v 1000 archivo  
carloslenovo@DESKTOP-VJJU969:~$ setfattr -n user.LXGID -v 1001 archivo  
carloslenovo@DESKTOP-VJJU969:~$
```

Figura 3.3.2: Añadir metadatos a un archivo de WSL

Para ver los atributos extendidos de un archivo en WSL, se puede usar el comando `getfattr` (instalar paquete `attr`). En este caso, para ver los valores de los atributos extendidos `user.LXUID`, `user.LXGID` y `user.LXMOD` del fichero anterior:



```
carloslenovo@DESKTOP-VJJU969: ~  
carloslenovo@DESKTOP-VJJU969:~$ getfattr -n user.LXGID archivo  
# file: archivo  
user.LXGID="1001"  
  
carloslenovo@DESKTOP-VJJU969:~$ getfattr -n user.LXUID archivo  
# file: archivo  
user.LXUID="1000"  
carloslenovo@DESKTOP-VJJU969:~$
```

Figura 3.3.3: Visualizar metadatos de un archivo en WSL

Un ejemplo de uso de atributos extendidos de NTFS en WSL podría ser trabajar con archivos de Windows en un entorno de desarrollo de Linux. Al habilitar los metadatos en WSL, se puede cambiar el propietario y los permisos de los archivos del proyecto como si se estuviera trabajando en un sistema de archivos nativo de Linux. Esto mejora la interoperabilidad entre Windows y Linux al trabajar con archivos compartidos entre ambos sistemas operativos. Por último, destacar que estos atributos extendidos solo afectan directamente a WSL para interpretar y aplicar correctamente los permisos de archivo de Linux en archivos de Windows cuando se accede a ellos desde WSL.

A continuación, se va a explicar los posibles escenarios que se pueden encontrar:

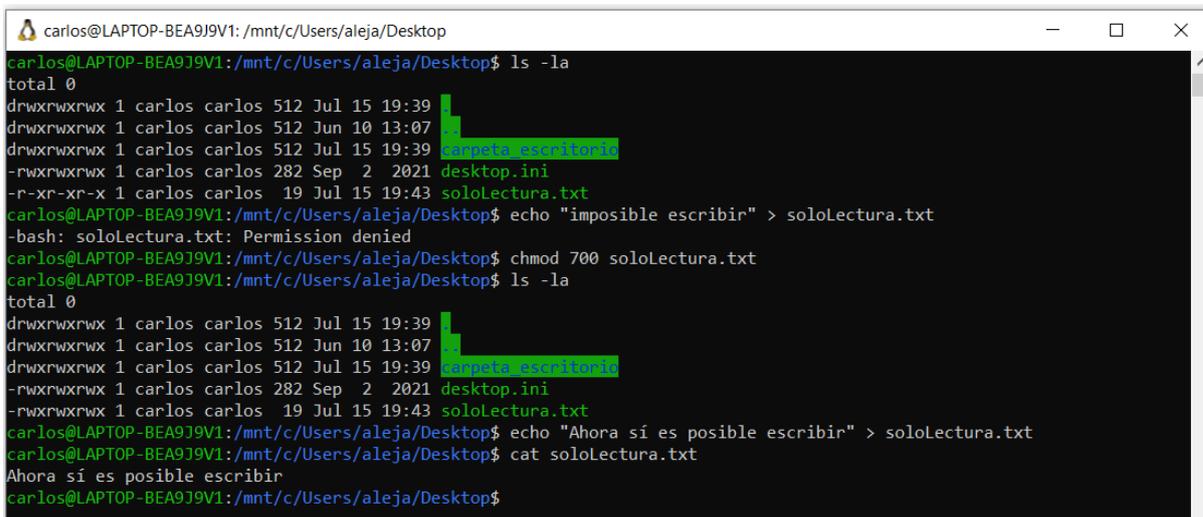
### 3.3.1. Acceso a archivos creados en Windows desde WSL

Cuando se accede desde WSL a archivos o directorios que se encuentran en `/mnt/c` (suponiendo que sea C la unidad del disco donde está instalado WSL).

En tal caso, los permisos de los ficheros que cuelguen de ese directorio dependerán de si han sido creados desde WSL y ya tienen metadatos o de si han sido creados desde Windows y no tienen metadatos.

En el caso donde el fichero o directorio no tiene metadatos asociados, se convierten los permisos que tiene el usuario de Windows en el sistema de permisos de Linux, es decir, se convierten los permisos en bits de lectura, escritura o ejecución. Estos valores se aplicarán tanto para el usuario, grupo y otros (en el sistema de permisos de Linux).

Ejemplo: si la cuenta de usuario de Windows tiene acceso de lectura y ejecución, pero no tiene acceso de escritura al archivo, en WSL los permisos se mostrarán como r-x para el usuario, el grupo y otros. Si el archivo cuenta con el atributo *Sólo lectura* establecido en Windows, no tendrá permisos de escritura en Linux. No obstante, estos permisos seguirán pudiendo ser modificados desde WSL.



```
carlos@LAPTOP-BEA9J9V1: /mnt/c/Users/aleja/Desktop
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$ ls -la
total 0
drwxrwxrwx 1 carlos carlos 512 Jul 15 19:39 [redacted]
drwxrwxrwx 1 carlos carlos 512 Jun 10 13:07 [redacted]
drwxrwxrwx 1 carlos carlos 512 Jul 15 19:39 carpeta_escritorio
-rwxrwxrwx 1 carlos carlos 282 Sep  2 2021 desktop.ini
-r-xr-xr-x 1 carlos carlos 19 Jul 15 19:43 soloLectura.txt
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$ echo "imposible escribir" > soloLectura.txt
-bash: soloLectura.txt: Permission denied
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$ chmod 700 soloLectura.txt
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$ ls -la
total 0
drwxrwxrwx 1 carlos carlos 512 Jul 15 19:39 [redacted]
drwxrwxrwx 1 carlos carlos 512 Jun 10 13:07 [redacted]
drwxrwxrwx 1 carlos carlos 512 Jul 15 19:39 carpeta_escritorio
-rwxrwxrwx 1 carlos carlos 282 Sep  2 2021 desktop.ini
-rwxrwxrwx 1 carlos carlos 19 Jul 15 19:43 soloLectura.txt
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$ echo "Ahora sí es posible escribir" > soloLectura.txt
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$ cat soloLectura.txt
Ahora sí es posible escribir
carlos@LAPTOP-BEA9J9V1:/mnt/c/Users/aleja/Desktop$
```

Figura 3.3.4: Cambio de permisos a ficheros creados en Windows desde WSL

Si se ejecutase `chmod 000 soloLectura.txt`, los permisos cambiarían al estado inicial cuando estaba establecido como “Sólo lectura” en Windows. Es decir, los permisos de dicho fichero figurarían como: `-r-xr-xr-x 1 carlos carlos 30 Jul 15 19:44 soloLectura.txt`

**Si el archivo tiene metadatos**, se usarán esos valores de metadatos en lugar de convertir los permisos vigentes del usuario de Windows.

### 3.3.2. Acceso a archivos creados en WSL desde Windows

Al crear un archivo en WSL, este se crea por defecto con unos permisos `-rw- r-- r--` Windows podrá realizar las acciones de lectura, escritura o ejecución de este archivo en función de los permisos que tenga asignados el usuario propietario de dicho fichero.

Por otra parte, estos permisos que han sido asignados desde WSL no pueden ser modificados desde Windows. Si el fichero no tuviera ningún tipo de permiso (`chmod 000`), no sería posible trabajar con ese fichero desde Windows hasta que se cambiaran los permisos del usuario propietario desde WSL.

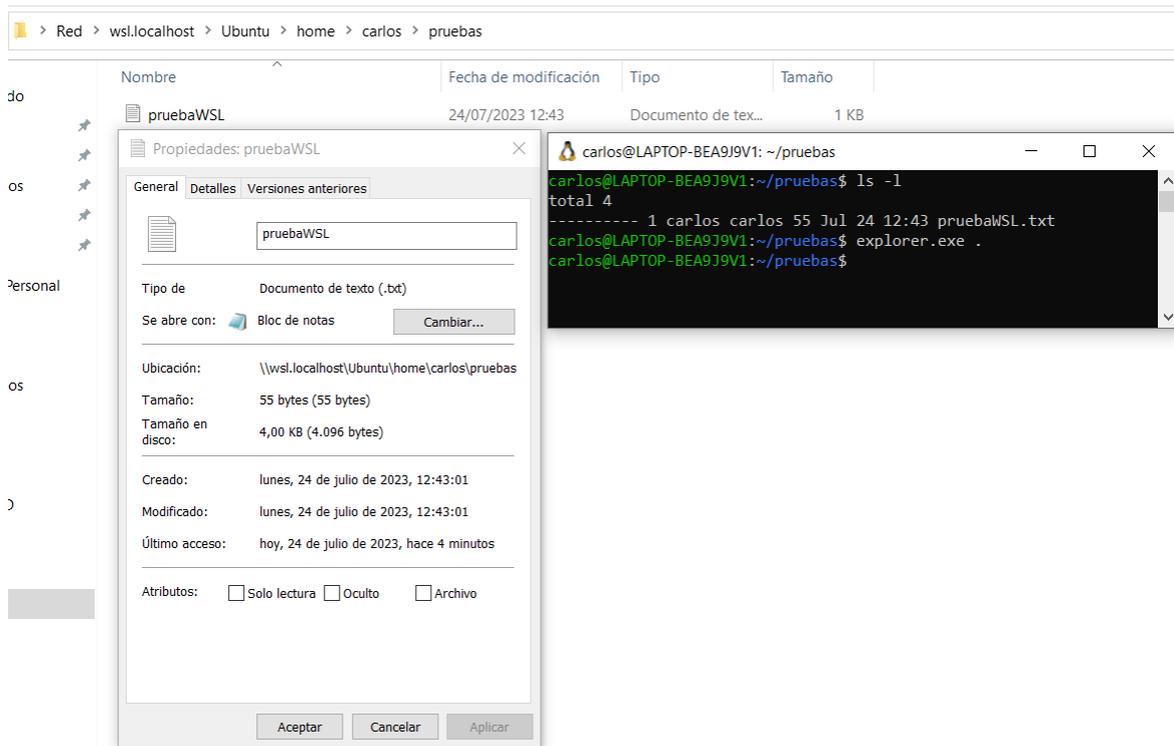


Figura 3.3.5: Acceso a fichero creado sin permisos en WSL desde Windows

Como se puede observar en la anterior imagen, en la pestaña de propiedades del fichero no aparece la sección de “seguridad” ya que es un fichero que ha sido creado desde WSL. Una vez cambiados los permisos, por ejemplo mediante el comando `chmod 700`, ese fichero sí que será accesible desde Windows.

### 3.4. Deshabilitación de la interoperabilidad

Aunque esta función puede deshabilitarse configurando los ficheros de configuración de WSL, los usuarios pueden deshabilitar la capacidad de ejecutar herramientas de Windows para una única sesión de WSL mediante la ejecución del siguiente comando como administrador:

```
echo 0 > /proc/sys/fs/binfmt_misc/WSLInterop
```

Para volver a habilitar los archivos binarios de Windows, es necesario salir de todas las sesiones de WSL y volver a ejecutar `bash.exe` o de manera alternativa, el siguiente comando como administrador:

```
echo 1 > /proc/sys/fs/binfmt_misc/WSLInterop
```

Sin embargo, esta deshabilitación de la interoperabilidad no se conservará entre sesiones de WSL: la interoperabilidad se habilitará de nuevo cuando se inicie una nueva sesión.



significativamente la interoperabilidad entre Windows y sistemas basados en Linux. En el caso particular de sshfs, el sistema de archivos montado sólo es accesible desde WSL.

```
carlos@LAPTOP-BEA9J9V1: ~/punto_montaje
carlos@LAPTOP-BEA9J9V1:~$ mkdir punto_montaje
carlos@LAPTOP-BEA9J9V1:~$ cd punto_montaje/
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$ ls
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$ sshfs carlosgp@f-13103-pc01.aulas.eif.unjc.es:/home/alumnos/carlosgp/Documentos
~/home/carlos/punto_montaje
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$ ls
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$ cd ..
carlos@LAPTOP-BEA9J9V1:~$ cd punto_montaje/
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$ ls
fichero_sshfs_montado.txt otro_ejemplo.txt
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$ cat otro_ejemplo.txt
Se puede editar el fichero remoto desde la máquina local!
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$ cd ..
carlos@LAPTOP-BEA9J9V1:~$ umount punto_montaje
carlos@LAPTOP-BEA9J9V1:~$ cd punto_montaje/
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$ ls
carlos@LAPTOP-BEA9J9V1:~/punto_montaje$
```

Figura 4.1.2: Montaje de directorio remoto vía sshfs

```
carlos@LAPTOP-BEA9J9V1: ~
carlos@LAPTOP-BEA9J9V1:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/carlos/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/carlos/.ssh/id_ed25519
Your public key has been saved in /home/carlos/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:jVS4noyixPxpBE4HAv38LNku0xQDkQaaJjs5wbYnIn8 carlos@LAPTOP-BEA9J9V1
The key's randomart image is:
+--[ED25519 256]--+
|+o.o  ..      |
|+. *   ..      |
|=* =   ..      |
|+++ *  ..o     |
|BB + B +S..   |
|o+B * = +     |
|..=E=         |
|..* .         |
|..+          |
+----[SHA256]-----+
carlos@LAPTOP-BEA9J9V1:~$
```

Figura 4.1.3: Comando ssh-keygen

```
carlos@LAPTOP-BEA9J9V1: ~
carlos@LAPTOP-BEA9J9V1:~$ pwd
/home/carlos
carlos@LAPTOP-BEA9J9V1:~$ ls
carlos@LAPTOP-BEA9J9V1:~$ scp -r carlosgp@f-13103-pc01.aulas.eif.unjc.es:/home/alumnos/carlosgp/Documentos .
prueba_scp.txt
100% 0 0.0KB/s 00:00
carlos@LAPTOP-BEA9J9V1:~$ ls
Documentos
carlos@LAPTOP-BEA9J9V1:~$ cd Documentos/
carlos@LAPTOP-BEA9J9V1:~/Documentos$ ls
prueba_scp.txt
carlos@LAPTOP-BEA9J9V1:~/Documentos$
```

Figura 4.1.4: Comando scp

#### 4.1.1. Conexión SSH desde máquina externa de la misma red a WSL

Si bien es cierto que es posible realizar conexiones SSH a la máquina local donde está instalado WSL (aunque esto carece de sentido ya que se va a realizar una conexión a la misma máquina desde la que se está conectando), para poder realizar conexiones SSH

desde una máquina externa a una máquina Windows que tiene instalado WSL, es necesario hacer una serie de configuraciones previas tanto en Windows como en WSL ya que este último no cuenta con este funcionamiento por defecto.

### 1. Instalar servidor de OpenSSH en WSL

```
apt install openssh-server
```

### 2. Editar configuración del fichero /etc/ssh/sshd\_config

Es necesario descomentar la siguiente información que, por defecto, está comentada:

```
Port 22 #DESCOMENTAR
ListenAddress 0.0.0.0 #DESCOMENTAR
PasswordAuthentication yes #DESCOMENTAR
```

### 3. Reiniciar demonio de SSH

Al mismo tiempo tras haberlo reiniciado, asegurarse de que el demonio de SSH esté corriendo en la máquina. También puede ser de utilidad hacer que el servicio de SSH se inicie automáticamente al arrancar la máquina.

```
sudo systemctl restart ssh && sudo systemctl enable ssh && sudo
systemctl status ssh
```

### 4. Obtener dirección IP de WSL

En el caso de estudio, la dirección IP es *172.31.228.108*

### 5. Redireccionar el puerto desde Windows a WSL

Para ello, desde el intérprete de comandos de Windows, ejecutar el comando `netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=<PUERTO> connectaddress=<IP_WSL> connectport=<PUERTO>`

En este caso concreto, el comando resultante quedaría de la siguiente manera:  
`netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=22 connectaddress=172.31.228.108 connectport=22`

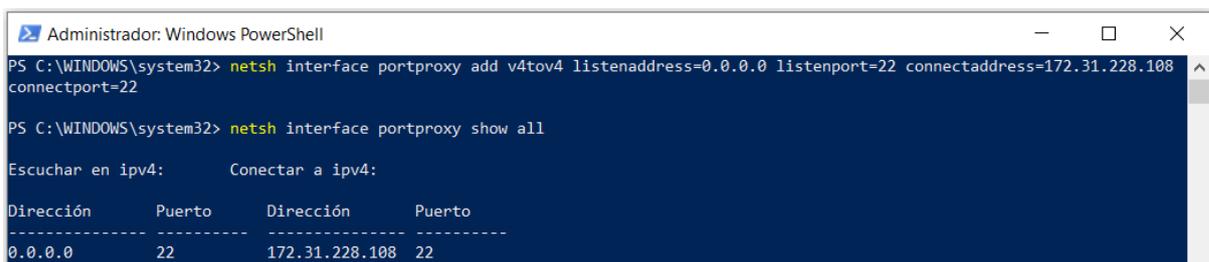


Figura 4.1.5: Redirección de puerto de SSH a WSL

### 6. Crear regla de conexión en el Firewall

Para ello, abrir la aplicación de Windows Defender *Firewall* ejecutando el comando `wf` desde el intérprete de comandos de Windows. Esto abrirá una sesión gráfica de dicha aplicación.

A continuación, seguir el orden que marca las flechas de la siguiente imagen:

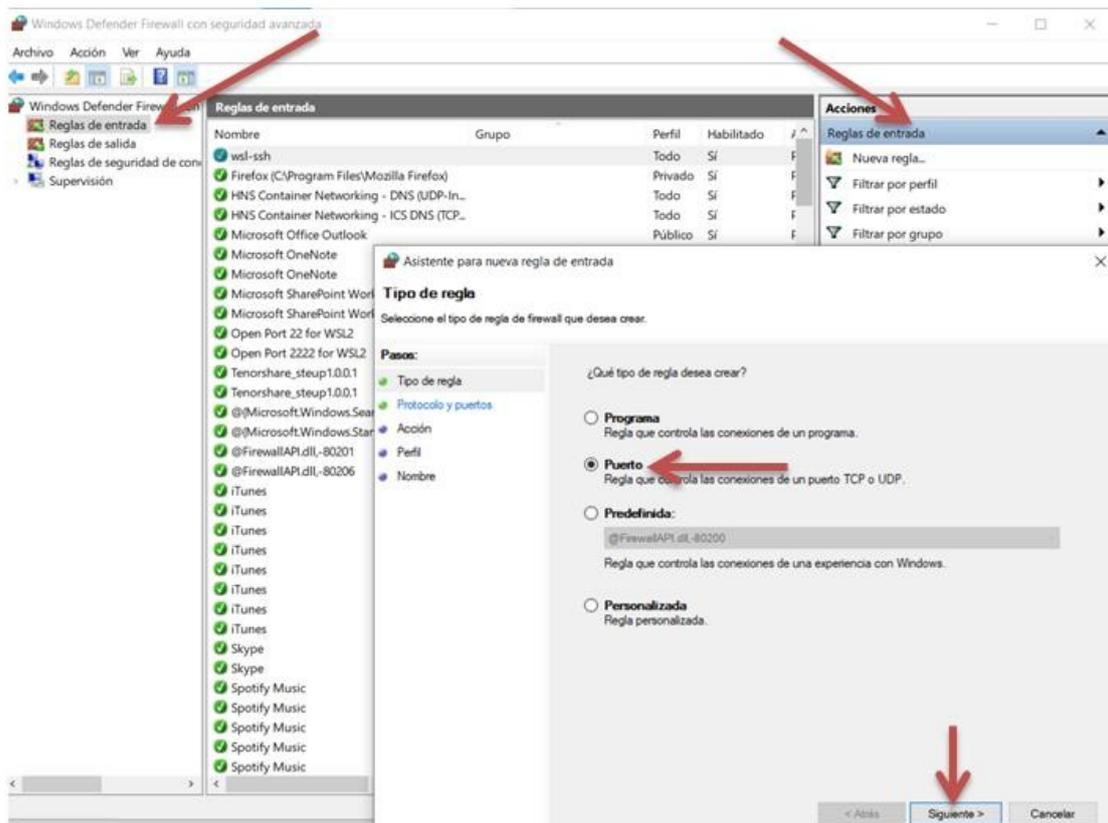


Figura 4.1.6: Configuración de regla de Firewall I

Tras haber clicado sobre el botón *Siguiente*, aparecerá la siguiente pestaña en la que habrá que seleccionar la opción de TCP y el puerto 22 como puerto específico.

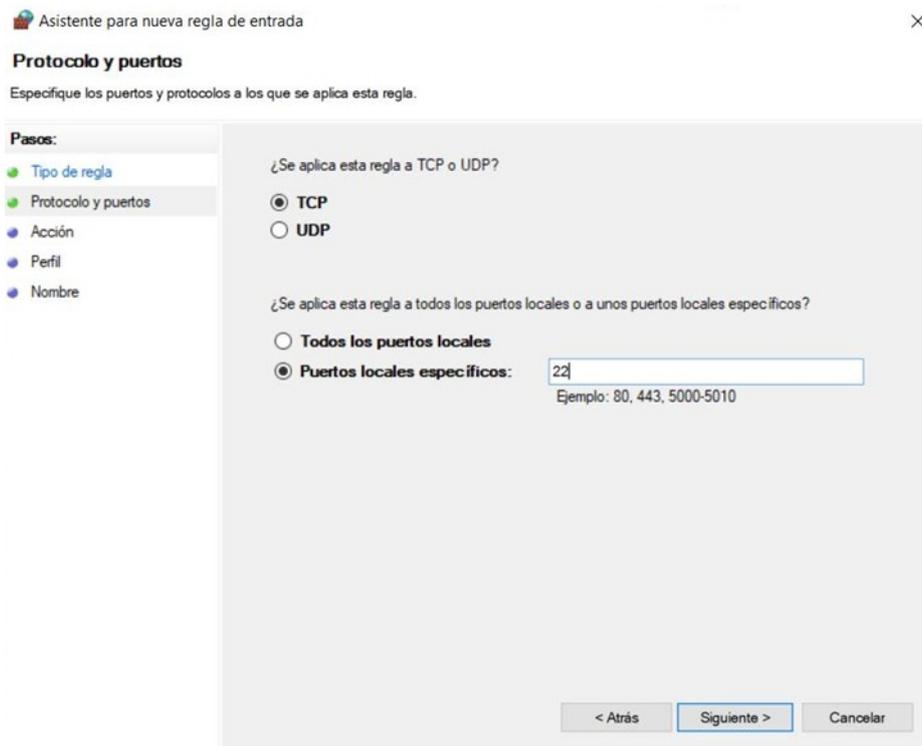


Figura 4.1.7: Configuración de regla de Firewall II

Tras clicar el botón *Siguiente*, aparecerá la siguiente pestaña:

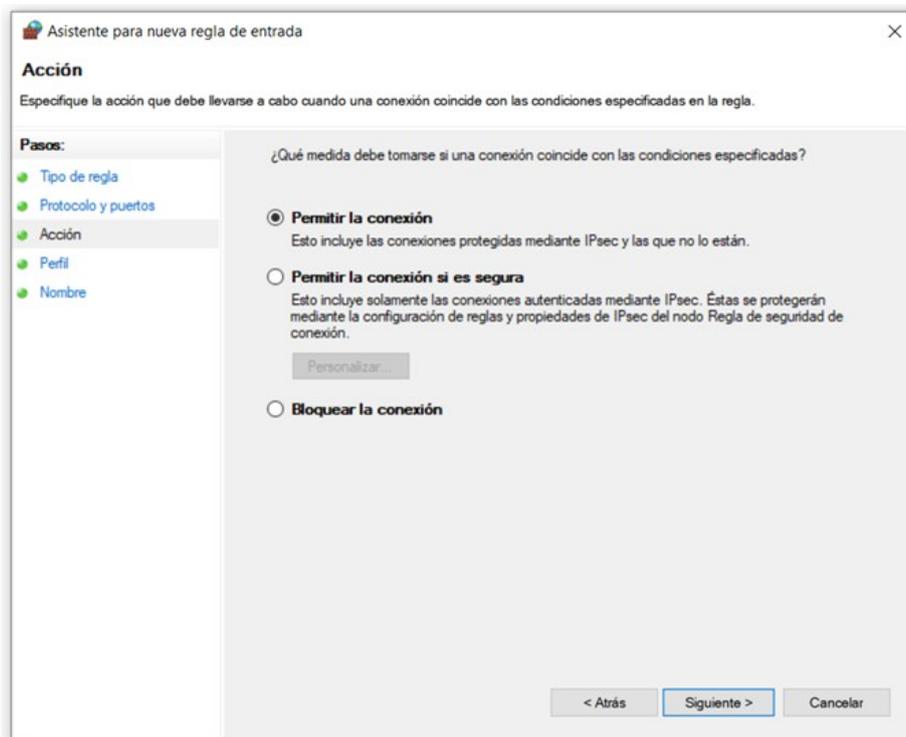


Figura 4.1.8: Configuración de regla de Firewall III

A continuación, marcar las tres opciones de la siguiente pestaña que se despliega tras clicar el botón *Siguiente*.

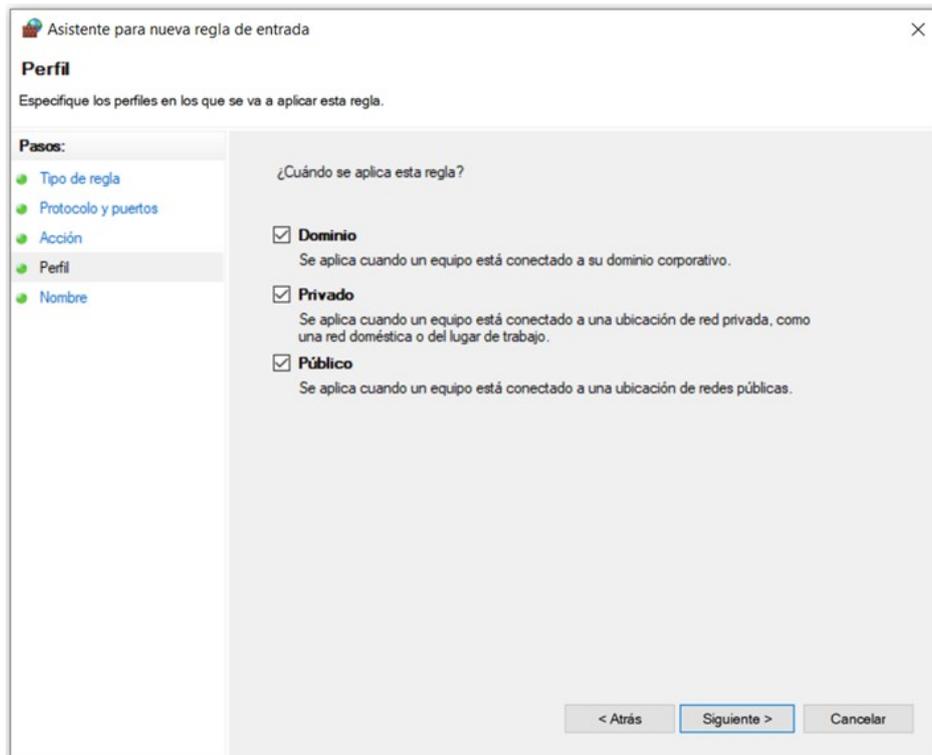


Figura 4.1.9: Configuración de regla de Firewall IV

A continuación, será necesario escribir un nombre que identifique al regla creada.

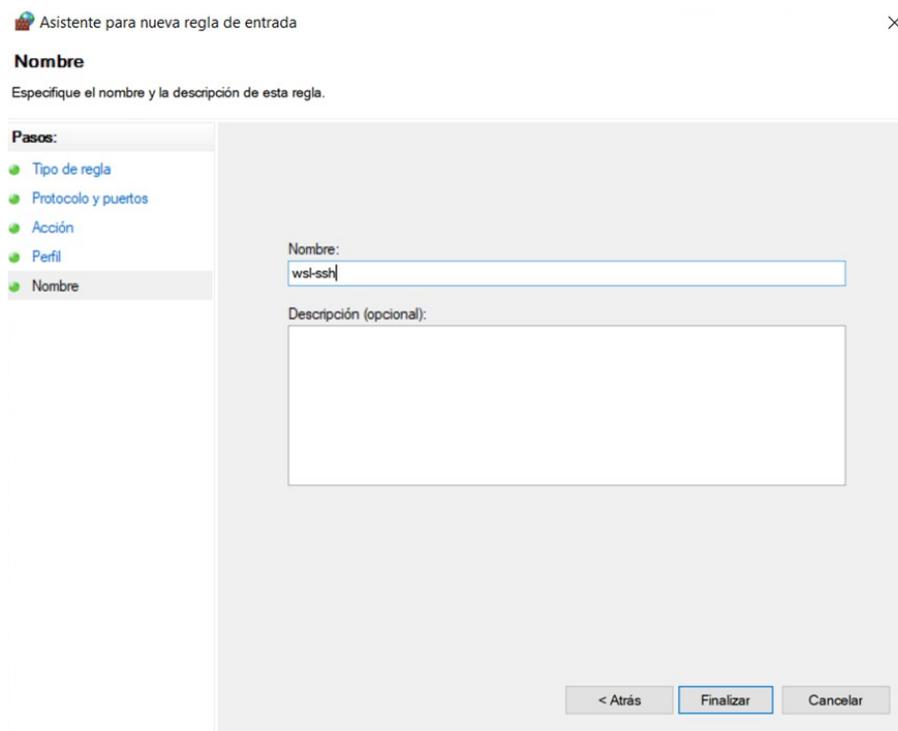
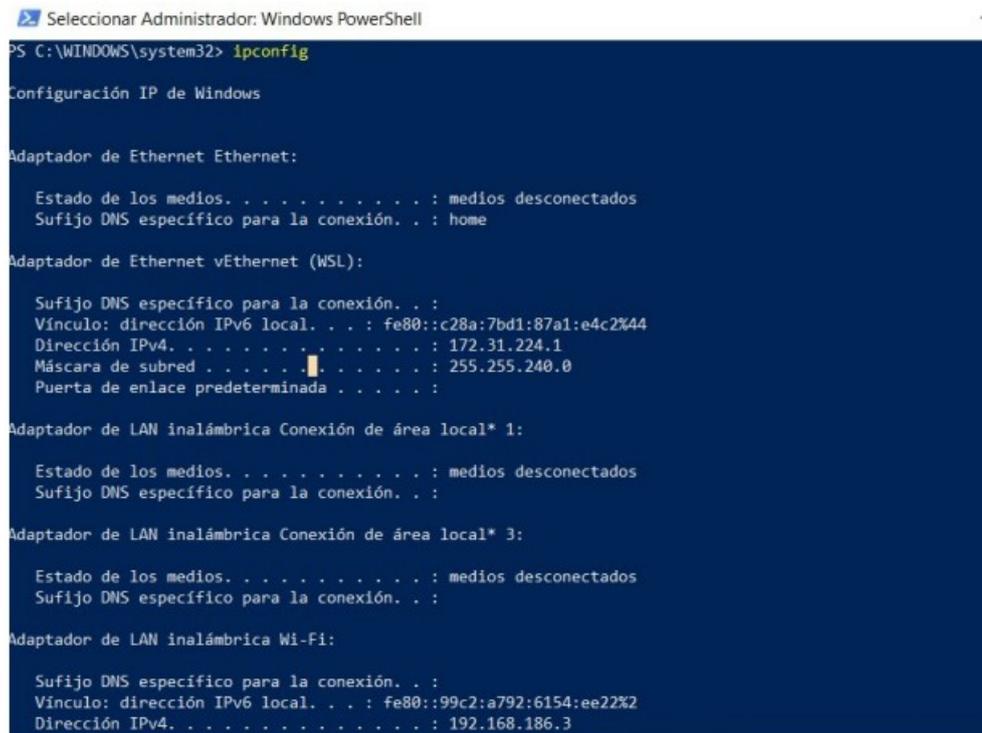


Figura 4.1.10: Configuración de regla de Firewall V

Por último, la dirección IP con la que hay que establecer la conexión es con la IP de la red a la que estén conectadas las máquinas. En este caso, la IP es 192.168.186.3.

Esta IP es dinámica. Por ello, antes de establecer la conexión SSH será necesario asegurarse de cuál es la IP correcta.



```
PS C:\WINDOWS\system32> ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . : home

Adaptador de Ethernet vEthernet (WSL):

    Sufijo DNS específico para la conexión. . :
    Vínculo: dirección IPv6 local. . . : fe80::c28a:7bd1:87a1:e4c2%44
    Dirección IPv4. . . . . : 172.31.224.1
    Máscara de subred. . . . . : 255.255.240.0
    Puerta de enlace predeterminada. . . . . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 3:

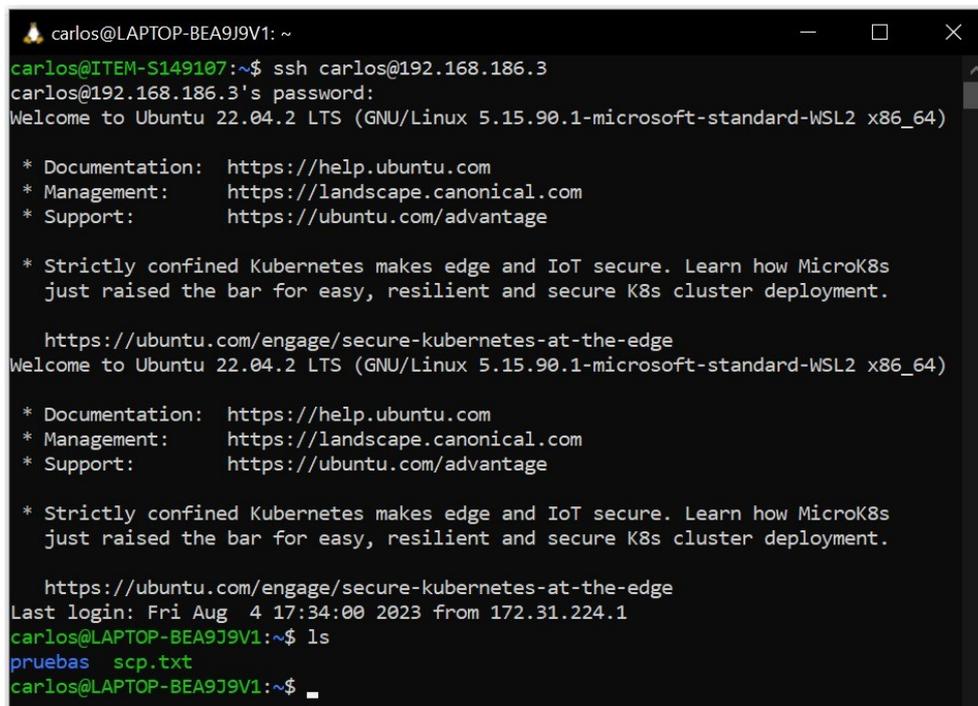
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . :
    Vínculo: dirección IPv6 local. . . : fe80::99c2:a792:6154:ee22%2
    Dirección IPv4. . . . . : 192.168.186.3
```

Figura 4.1.11: Averiguar IP de la conexión de red

Tras haber realizado estas configuraciones, será posible acceder vía SSH a dicha máquina desde una máquina externa. A continuación, se muestra un ejemplo de acceso de una máquina Windows con WSL a otra máquina con las mismas condiciones.



```
carlos@LAPTOP-BEA9J9V1: ~
carlos@ITEM-S149107:~$ ssh carlos@192.168.186.3
carlos@192.168.186.3's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

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

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

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

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge
Last login: Fri Aug  4 17:34:00 2023 from 172.31.224.1
carlos@LAPTOP-BEA9J9V1:~$ ls
pruebas  scp.txt
carlos@LAPTOP-BEA9J9V1:~$
```

Figura 4.1.12: Conexión SSH establecida desde máquina externa

Para establecer conexiones SSH con máquinas ubicadas en redes externas, se requiere efectuar ajustes en los puertos del router. Para llevar a cabo esta acción, se recomienda hacer referencia a la guía proporcionada por el fabricante del router para una configuración adecuada.

## 4.2. Virtualización: contenedores

WSL permite utilizar Docker y Podman de manera nativa en Windows. Estas herramientas son ampliamente utilizadas para la creación, gestión y ejecución de contenedores.

Con WSL, los usuarios pueden construir imágenes de contenedores, ejecutar y gestionar contenedores, así como aprovechar las redes y volúmenes definidos en Docker y Podman. Esto permite una integración fluida con el ecosistema de contenedores existente y una mayor compatibilidad entre Windows y Linux en términos de desarrollo y despliegue de aplicaciones.

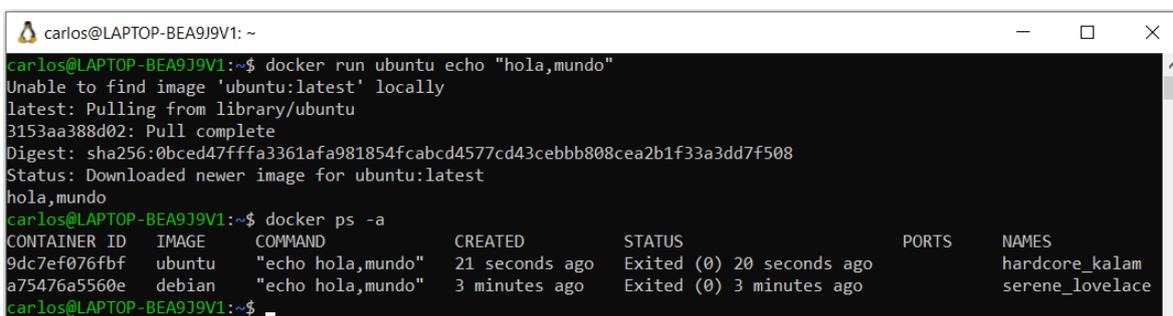
### 4.2.1. Docker

Docker es una plataforma de virtualización a nivel de sistema operativo que se lanzó a principios del año 2013 que permite empaquetar aplicaciones y sus dependencias en contenedores ligeros y portátiles. Estos contenedores se ejecutan de manera aislada, lo que facilita la implementación y el despliegue de aplicaciones en diferentes entornos, sin importar las diferencias en la infraestructura subyacente. Docker simplifica el proceso de desarrollo, pruebas y despliegue de aplicaciones al ofrecer una solución eficiente y consistente para la gestión de entornos de ejecución.

Esta plataforma de virtualización no viene instalada por defecto por lo que es necesario instalarla previamente para poder utilizarla (`apt install docker.io`).

Para poder trabajar con Docker, es necesario hacer una serie de configuraciones previas:

- Crear grupo docker: `sudo addgroup docker`
- Añadir el usuario al grupo docker: `sudo adduser $USER docker`



```
carlos@LAPTOP-BEA9J9V1: ~$ docker run ubuntu echo "hola,mundo"
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
3153aa388d02: Pull complete
Digest: sha256:0bced47fffa3361afa981854fcabcd4577cd43cebbb808cea2b1f33a3dd7f508
Status: Downloaded newer image for ubuntu:latest
hola,mundo
carlos@LAPTOP-BEA9J9V1:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
9dc7ef076fbf   ubuntu   "echo hola,mundo"       21 seconds ago Exited (0) 20 seconds ago             hardcore_kalam
a75476a5560e   debian  "echo hola,mundo"       3 minutes ago  Exited (0) 3 minutes ago             serene_lovelace
carlos@LAPTOP-BEA9J9V1:~$
```

Figura 4.2.1: Prueba de tipo "hola, mundo" en Docker

A continuación, se muestra un contenedor de prueba de Docker donde al ponerse en marcha, el usuario *root* tiene una sesión de *shell* abierta y que, además, tiene disponible en el alias *h* el comando `echo "hola, mundo"`.

```
root@carlalias01: /
carlos@LAPTOP-BEA9J9V1:~/pruebasDocker$ ls
construye.sh context lanza_carlalias01.sh
carlos@LAPTOP-BEA9J9V1:~/pruebasDocker$ cd context/
carlos@LAPTOP-BEA9J9V1:~/pruebasDocker/context$ ls
Dockerfile bash_aliases entrypoint.sh
carlos@LAPTOP-BEA9J9V1:~/pruebasDocker/context$ cd ..
carlos@LAPTOP-BEA9J9V1:~/pruebasDocker$ ./construye.sh
Sending build context to Docker daemon 4.096kB
Step 1/5 : FROM ubuntu:22.04
--> 5a81c4b8502e
Step 2/5 : RUN apt-get update -y && apt-get upgrade -y
--> Using cache
--> 05989c9ce187
Step 3/5 : COPY bash_aliases /root/.bash_aliases
--> Using cache
--> 5c35788375f4
Step 4/5 : COPY entrypoint.sh /
--> Using cache
--> 83ae592e9bd0
Step 5/5 : ENTRYPOINT ["/entrypoint.sh"]
--> Using cache
--> d3a14cbef495
Successfully built d3a14cbef495
Successfully tagged alias:latest
carlos@LAPTOP-BEA9J9V1:~/pruebasDocker$ ./lanza_carlalias01.sh
root@carlalias01:/# h
hola, mundo
root@carlalias01:/# _
```

Figura 4.2.2: Prueba contenedor Docker

## 4.2.2. Podman

Podman es un proyecto más nuevo que Docker que se lanzó a finales del año 2017. Es una herramienta de virtualización similar a Docker que permite a los usuarios crear, ejecutar y administrar contenedores sin necesidad de un demonio en segundo plano. Aunque está menos extendido que Docker, actualmente su uso está aumentando mucho. Proporciona una experiencia de virtualización segura y portátil, permitiendo la gestión de contenedores sin requerir privilegios de usuario root. Además, su *interface* es prácticamente idéntico al de Docker, con la que la transición de uno a otro, o el uso en paralelo, es casi inmediato.

Para comprobar su correcto funcionamiento en WSL, se va a crear y ejecutar un contenedor de prueba idéntico al de la figura *Figura 4.2.2: Prueba contenedor Docker*.

```
root@carlalias01: /
carlos@LAPTOP-BEA9J9V1:~/pruebasPodman$ ./construye.sh
STEP 1/5: FROM ubuntu:22.04
STEP 2/5: RUN apt-get update -y && apt-get upgrade -y
--> Using cache def09324f283bf5bbac0123d71965543f1ed06ef8b890443e2b0ae2f8eadc4e9
--> def09324f28
STEP 3/5: COPY bash_aliases /root/.bash_aliases
--> Using cache f0bc587509531813cae19946d94ac6f13ec985705987c92228a5da2b80a131e8
--> f0bc5875095
STEP 4/5: COPY entrypoint.sh /
--> Using cache 13825366d80b3cd4ab0e45e5e06ebbea06b771b16a20958f8abff6e3b2093d22
--> 13825366d80
STEP 5/5: ENTRYPOINT ["/entrypoint.sh"]
--> Using cache 321b462a49300ae67033edfadc9acfc38afde35ecb053f63fbf755a07063f589
COMMIT alias
--> 321b462a493
Successfully tagged localhost/alias:latest
321b462a49300ae67033edfadc9acfc38afde35ecb053f63fbf755a07063f589
carlos@LAPTOP-BEA9J9V1:~/pruebasPodman$ ./lanza_carlalias01.sh
root@carlalias01:/# h
hola, mundo
root@carlalias01:/#
```

Figura 4.2.3: Prueba contenedor Podman

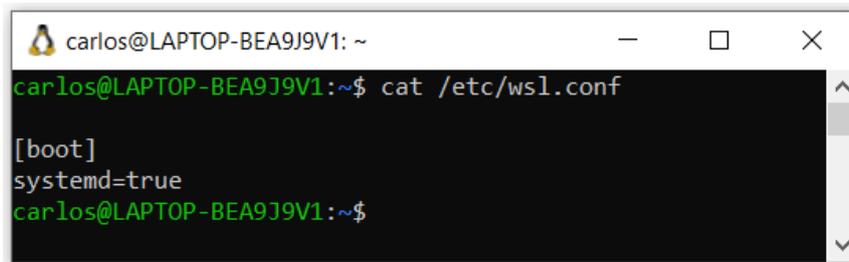
## 5. Archivos de configuración de WSL

En WSL, existen dos archivos de configuración principales: `wsl.conf` y `.wslconfig`. Estos archivos permiten personalizar y ajustar diferentes aspectos del entorno de WSL. El fichero `.wslconfig` sirve para configurar las opciones globalmente en todas las distribuciones instaladas que se ejecutan en WSL 2. Por el contrario, el fichero `wsl.conf` sirve para configurar las opciones por distribución para distribuciones de Linux que se ejecutan en WSL.

### 5.1. `wsl.conf`

Este archivo de configuración se puede encontrar en el directorio `/etc` de WSL. Se usa para configurar las opciones por distribución. Los valores configurados en este archivo sólo se aplicarán a la distribución específica de Linux que contiene el directorio donde se almacena este archivo.

Por defecto, el contenido de este fichero es el siguiente:

A screenshot of a terminal window titled 'carlos@LAPTOP-BEA9J9V1: ~'. The terminal shows the command 'cat /etc/wsl.conf' being executed. The output of the command is displayed in green text on a black background: '[boot]' followed by 'systemd=true' on the next line. The prompt 'carlos@LAPTOP-BEA9J9V1:~\$' is visible at the end of the output.

```
carlos@LAPTOP-BEA9J9V1:~$ cat /etc/wsl.conf
[boot]
systemd=true
carlos@LAPTOP-BEA9J9V1:~$
```

Figura 5.1.1: Contenido del fichero `wsl.conf`

El archivo `wsl.conf` utiliza una sintaxis simple basada en secciones y opciones. Cada sección se define entre corchetes (`[]`) y dentro de cada sección se especifican las opciones y sus valores.

En la imagen anterior, el contenido que se muestra es el siguiente:

- `[boot]`: Esta sección define opciones relacionadas con el proceso de inicio de la distribución de Linux en WSL.
- `systemd=true`: Esta opción indica que se debe utilizar `systemd` como el proceso de inicio principal dentro de la distribución de Linux en WSL. `Systemd` es un sistema de inicialización y administración de servicios ampliamente utilizado en distribuciones Linux modernas.

Cuando se realicen cambios en este fichero, es necesario reiniciar WSL, bien cerrándolo y abriéndolo, bien ejecutando el comando `wsl.exe --shutdown` desde el intérprete de comandos de Windows. Los cambios tardan alrededor de 8 segundos en aplicarse desde que WSL se ha cerrado.

### 5.1.1. Secciones configurables disponibles

Aunque ciertas tareas se pueden realizar ejecutando comandos de WSL en el intérprete de comandos de Windows, también es posible hacerlo de manera más rápida y automatizada añadiendo líneas de configuración en el fichero *wsl.conf*.

El archivo *wsl.conf* admite cinco secciones configurables: *[automount]*, *[network]*, *[interop]*, *[user]* y *[boot]*.

- **[automount]:** esta sección se utiliza para controlar el montaje automático de unidades de WSL. Al establecer esta etiqueta en el archivo *wsl.conf*, se puede definir cómo se manejarán las unidades de Windows al acceder a ellas desde la distribución de Linux en WSL.

Las opciones con las que cuenta son las siguientes:

*enabled = (boolean):* Por defecto, está inicializado a *true*. Esta opción hace que las unidades fijas (por ejemplo, *C:/*) se monten automáticamente en el directorio */mnt* de WSL. Si su valor fuera *false*, significa que las unidades no serán montadas de manera automática.

*mountFsTab = (boolean):* Por defecto, está inicializado a *true*. Esta opción sirve para se procese en cada arranque de WSL el fichero */etc/fstab*. En este archivo se pueden declarar otros sistemas de archivos para montarlos automáticamente en WSL.

*root = (string):* Establece el directorio donde se montarán automáticamente las unidades fijas. Por defecto, su valor es */mnt/* por lo que la unidad C del sistema de archivos de Windows se monta en */mnt/c/*. Si se cambia */mnt/* a otra ruta, la unidad será montada en esa nueva ruta.

*options = (lista separada por comas):* Las opciones de montaje automático se aplican como opciones de montaje para todas las unidades montadas automáticamente. Si solamente se quisiera configurar una unidad específica, habría que configurarlo dentro del fichero */etc/fstab*.

Las opciones configurables dentro de esta sección son las siguientes:

- *uid:* es el identificador de usuario que se usa para el propietario de todos los archivos. Su valor por defecto es 1000.
- *gid:* es el identificador de grupo que se usa para el propietario de todos los archivos. Su valor por defecto es 1000.
- *umask:* es la máscara octal de permisos que se van a excluir para todos los archivos y directorios. Su valor por defecto es 000.
- *fmask:* es la máscara octal de permisos que se van a excluir para todos los archivos. Su valor por defecto es 000.
- *dmask:* es la máscara octal de permisos que se van a excluir para todos los directorios. Su valor por defecto es 000.

- *metadata*: indica si se agregan metadatos a archivos de Windows para admitir permisos del sistema Linux. Por defecto, esta opción está deshabilitada.
- *case*: especifica el comportamiento predeterminado de sensibilidad a mayúsculas y minúsculas al montar la unidad. Esta opción solo puede tomar dos valores: *off*, *dir* y *force*. Cuando se establece en *off*, los nuevos directorios creados en la unidad montada no serán sensibles a mayúsculas y minúsculas por defecto. Cuando se establece en *dir*, los nuevos directorios creados en la unidad montada serán sensibles a mayúsculas y minúsculas por defecto. Cuando se establece en *force*, todos los directorios, incluidos los existentes, se tratan como si fueran sensibles a mayúsculas y minúsculas.

- **[network]**: Esta sección se utiliza para realizar configuraciones de red. Las opciones con las que cuenta son las siguientes:

*generateHosts = (boolean)*: Por defecto, está inicializado a *true*. Cuando esta opción está habilitada, WSL generará una entrada en el archivo */etc/hosts* para cada dirección IP asignada a la máquina host de Windows. Esto facilita el acceso a recursos y servicios de red de la máquina host desde WSL utilizando nombres de host en lugar de direcciones IP.

*generateResolvConf = (boolean)*: Por defecto, está inicializado a *true*. Cuando esta opción está habilitada, WSL generará dinámicamente el archivo */etc/resolv.conf* cada vez que se inicie la distribución de Linux en WSL. Este archivo es utilizado por el sistema operativo Linux para resolver nombres de dominio a direcciones IP utilizando servidores DNS.

Al generar automáticamente el archivo */etc/resolv.conf*, WSL configura los servidores DNS de la máquina host de Windows para que sean utilizados por la distribución de Linux en WSL. Esto permite que la distribución de Linux en WSL utilice los mismos servidores DNS que la máquina host de Windows.

*hostname = (string)*: Su valor por defecto es el nombre de host de Windows. Esta opción establece el nombre de host que se usará para la distribución de WSL.

- **[interop]**: Esta sección se utiliza para realizar configuraciones sobre la interoperabilidad de WSL con Windows. Las opciones con las que cuenta son las siguientes:

*enabled = (boolean)*. Por defecto, está inicializado a *true*. La configuración de esta opción determinará si WSL admitirá el inicio de procesos de Windows o no.

*AppendWindowsPath = (boolean)*. Por defecto, está inicializado a *true*. Establecer esta opción determinará si WSL agregará elementos de ruta de acceso de Windows a la variable de entorno *\$PATH* de WSL.

- **[user]**: Esta sección se utiliza para realizar configuraciones sobre los usuarios de la distribución de WSL. La opción con la que cuenta es la siguiente:

*default = (string)*. Por defecto, su valor es el nombre de usuario creado en la primera ejecución de WSL. Si se cambiara su valor a otro nombre de usuario perteneciente a la distribución donde se está trabajando, ese será el usuario que iniciará la sesión de WSL.

- **[boot]:** Esta sección sólo está disponible en Windows 11 y Windows Server 2022. Sirve para realizar configuraciones de arranque de WSL. La opción con la que cuenta es la siguiente:

*command = (string)*. Siendo *command* el comando que se quiera ejecutar en cada arranque de WSL. Por defecto, su valor es "", es decir, ningún comando se ejecutará por defecto al iniciar WSL.

Un ejemplo de configuración del archivo `wsl.conf` podría ser el siguiente que se muestra en esta imagen extraída de la página oficial de Microsoft:

```
Bash

# Automatically mount Windows drive when the distribution is launched
[automount]

# Set to true will automount fixed drives (C:/ or D:/) with DrvFs under the
root directory set above. Set to false means drives won't be mounted
automatically, but need to be mounted manually or with fstab.
enabled = true

# Sets the directory where fixed drives will be automatically mounted. This
example changes the mount location, so your C-drive would be /c, rather than
the default /mnt/c.
root = /

# DrvFs-specific options can be specified.
options = "metadata,uid=1003,gid=1003,umask=077,fmask=11,case=off"

# Sets the `/etc/fstab` file to be processed when a WSL distribution is
launched.
mountFstab = true

# Network host settings that enable the DNS server used by WSL 2. This
example changes the hostname, sets generateHosts to false, preventing WSL
from the default behavior of auto-generating /etc/hosts, and sets
generateResolvConf to false, preventing WSL from auto-generating
/etc/resolv.conf, so that you can create your own (ie. nameserver 1.1.1.1).
[network]
hostname = DemoHost
generateHosts = false
generateResolvConf = false

# Set whether WSL supports interop process like launching Windows apps and
adding path variables. Setting these to false will block the launch of
Windows processes and block adding $PATH environment variables.
[interop]
enabled = false
appendWindowsPath = false

# Set the user when launching a distribution with WSL.
[user]
default = DemoUser

# Set a command to run when a new WSL instance launches. This example starts
the Docker container service.
[boot]
command = service docker start
```

Figura 5.1.2: Ejemplo de configuración del fichero `wsl.conf`

## 5.2. `.wslconfig`

Por defecto, este fichero no se crea directamente tras haber instalado WSL. Como este archivo no está creado, se aplicarán los valores predeterminados para WSL por lo que si se quiere realizar algún cambio de configuración, será necesario crearlo para sustituir la configuración predeterminada de WSL. Si se creara el fichero y tuviera un formato de marcado incorrecto, WSL iniciará de forma normal sin aplicar las opciones de configuración especificadas en este fichero.

La ruta donde ha de estar ubicado este fichero de configuración (en Windows) es `C:\Users\<UserName>`

### 5.2.1. Secciones configurables disponibles

El fichero `.wslconfig` usa la misma sintaxis que el fichero `wsl.conf` y cuenta con una única sección configurable que y varias opciones: `[wsl2]`.

- **[wsl2]:** Esta sección se utiliza para configurar opciones específicas de WSL2. Algunas de las opciones con las que cuenta solamente están disponible en Windows 11. Algunas de ellas son:

#### **Opciones disponibles en Windows 10 y posteriores:**

`kernel = (PATH)`. Representa la ruta de acceso absoluta de Windows a un `kernel` de Linux personalizado. Su valor por defecto es la bandeja de entrada proporcionada por el `kernel` compilado de Microsoft.

`memory = (numeroGB)`. Representa la cantidad de memoria que se le va a asignar a la máquina virtual de WSL2. Su valor general por defecto es el 50% de memoria total en Windows u 8 GB, lo que sea menor. Este valor dependerá del valor de la compilación de Windows.

`processors = (numero)`. Representa el número de procesadores lógicos que se van a asignar a la máquina virtual WSL. Su valor por defecto es el mismo número de procesadores lógicos que en Windows.

`localhostForwarding = (boolean)`: Esta opción se refiere a la capacidad de reenvío de puertos desde la máquina virtual de WSL al host de Windows a través de `localhost:puerto`. Cuando `localhostForwarding` está configurado en `true`, los puertos enlazados a `localhost` o `0.0.0.0` dentro de la máquina virtual de WSL pueden ser accedidos desde el host de Windows a través de `localhost:puerto`. Esto permite que las aplicaciones y servicios que se ejecutan en WSL sean accesibles desde el navegador web o desde otras aplicaciones en el host de Windows. Su valor por defecto está inicializado a `true`.

`kernelCommandLine = (string)`: Esta opción se refiere a los argumentos adicionales que se pueden especificar en la línea de comandos del `kernel` al iniciar una distribución de Linux en WSL. Estos argumentos permiten personalizar y configurar el comportamiento del `kernel` de Linux dentro de WSL.

Al establecer *kernelCommandLine* en una cadena de texto, se pueden proporcionar argumentos específicos que se pasen al *kernel* de Linux al iniciar la distribución. Estos argumentos pueden afectar diferentes aspectos del *kernel*, como la configuración del sistema, el comportamiento de los controladores de dispositivos, las opciones de red, entre otros. Su valor por defecto tiene asignado una cadena vacía.

*swap = (numeroGB)*: Esta opción se refiere al tamaño del archivo de intercambio (swap) que se agrega a la máquina virtual de WSL. El archivo de intercambio se utiliza como una extensión de la memoria RAM para manejar situaciones en las que la demanda de memoria supera el límite físico en el dispositivo de hardware.

Por defecto, el tamaño del archivo de intercambio se establece en el 25% del tamaño de la memoria en Windows, redondeado al valor más cercano en gigabytes (GB).

### **Opciones disponibles en Windows 11:**

*Safemode = (boolean)*: Esta opción inicia en modo seguro y deshabilita muchas de las características. Su valor por defecto es *false*.

*guiApplications = (boolean)*: Esta opción sirve para activar o desactivar la compatibilidad con aplicaciones de GUI en WSL. Su valor por defecto es *true*.

*debugConsole = (boolean)*: Esta opción sirve para para activar una ventana de consola de salida que muestra el contenido del comando *dmesg* al inicio de una instancia de distribución de WSL.

*vmIdleTimeout = (número)*: Número de milisegundos que una máquina virtual está inactiva antes de apagarla. Su valor por defecto es 60000.

Un ejemplo de configuración del archivo *.wslconfig* podría ser el siguiente que se muestra en esta imagen extraída de la página oficial de Microsoft:

```
Bash

# Settings apply across all Linux distros running on WSL 2
[wsl2]

# Limits VM memory to use no more than 4 GB, this can be set as whole
numbers using GB or MB
memory=4GB

# Sets the VM to use two virtual processors
processors=2

# Specify a custom Linux kernel to use with your installed distros. The
default kernel used can be found at https://github.com/microsoft/WSL2-Linux-
Kernel
kernel=C:\\temp\\myCustomKernel

# Sets additional kernel parameters, in this case enabling older Linux base
images such as Centos 6
kernelCommandLine = vsyscall=emulate

# Sets amount of swap storage space to 8GB, default is 25% of available RAM
swap=8GB

# Sets swapfile path location, default is
%USERPROFILE%\AppData\Local\Temp\swap.vhdx
swapfile=C:\\temp\\wsl-swap.vhdx

# Disable page reporting so WSL retains all allocated memory claimed from
Windows and releases none back when free
pageReporting=false

# Turn off default connection to bind WSL 2 localhost to Windows localhost
localhostforwarding=true

# Disables nested virtualization
nestedVirtualization=false

# Turns on output console showing contents of dmesg when opening a WSL 2
distro for debugging
debugConsole=true
```

*Figura 5.2.1: Ejemplo de configuración del fichero .wslconfig*

## 6. Conexión de dispositivos USB

La compatibilidad con la conexión de dispositivos USB, en el momento de escribir este manual (año 2023), no está disponible de forma nativa en WSL. Actualmente, la página oficial de Microsoft ofrece la posibilidad de conectar un dispositivo USB a una distribución de Linux que se ejecuta en WSL instalando el proyecto de código abierto *usbipd-win*. No obstante, esta solución no es, en el momento de redacción de este manual, todavía funcional por lo que el acceso al contenido del USB no es posible. Algunos usuarios de internet han intentado ponerle solución a este problema, por ejemplo, a través de modificaciones del *kernel* de Linux. Sin embargo, esta solución con el tiempo es muy probable que deje de funcionar en cuanto aparezcan nuevas versiones del núcleo.

Por otra parte, WSL sí que ofrece de manera nativa la posibilidad de acceder desde Windows a un disco externo formateado en ext4.

### 6.1. *usbipd-win*

#### 6.1.1. Requisitos previos

- Trabajar con una versión de Windows 11.
- Se requiere una máquina con un procesador x64/x86 ya que Arm64 no se admite actualmente con *usbipd-win*.
- Contar con una distribución de Linux instalada y establecida en WSL.
- Ejecutar un *kernel* de Linux 5.10.60.1 o posterior.

#### 6.1.2. ¿Qué es USB/IP?

USB/IP es un protocolo y una tecnología que permite compartir dispositivos USB a través de una red IP. A su vez, permite acceder y utilizar dispositivos USB conectados a un servidor desde otros equipos en la red, como si estuvieran conectados directamente a esos equipos locales.

*usbipd-win* es una implementación de software para Windows que actúa como un servidor USB/IP. Proporciona la funcionalidad necesaria para compartir dispositivos USB conectados a una computadora con otros equipos en la red. Esto permite que los dispositivos USB, como impresoras, escáneres o dispositivos de almacenamiento, sean accesibles y utilizables desde otros equipos sin tener que conectarlos físicamente a cada uno de ellos.

Para descargarlo, clicar en el enlace [Releases · dorssel/usbipd-win \(github.com\)](https://github.com/dorssel/usbipd-win/releases) para descargar e instalar en Windows el fichero con nombre *usbipd-win\_3.0.0.msi*.

### 6.1.3. Configuración

Una vez instalado el proyecto USB/IP, hay que instalar las herramientas de espacio de usuario y una base de datos de identificadores de hardware USB.

En este ejemplo, la distribución de Linux sobre la que se va a realizar el proceso de configuración es Ubuntu. Si se quisiera hacer lo mismo en otras distribuciones, puede ser necesario instalar un paquete de cliente usbip diferente.

Para continuar el proceso de configuración (distribución Ubuntu), es necesario ejecutar los dos siguientes comandos en WSL para que Windows pueda ejecutar un servicio que permita compartir dispositivos USB y las herramientas necesarias para que WSL pueda conectarse a los dispositivos compartidos:

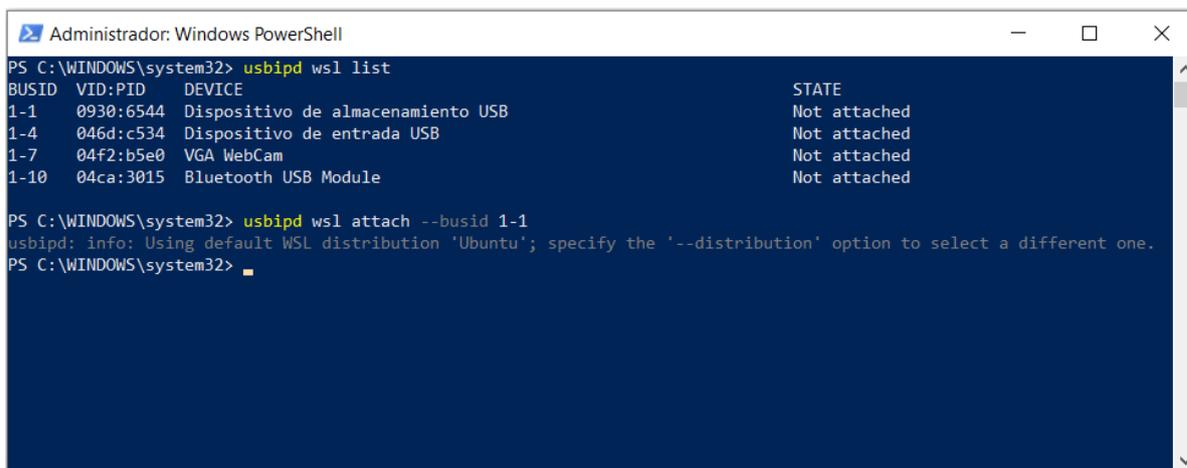
- `sudo apt install linux-tools-generic hwdata`
- `sudo update-alternatives --install /usr/local/bin/usbip usbip /usr/lib/linux-tools/*-generic/usbip 20`

### 6.1.4. Comprobación

En primer lugar, conectar el dispositivo USB al puerto correspondiente de la máquina. Tras ello, comprobar que, efectivamente, el sistema detecta dicho dispositivo; para ello, ejecutar desde el intérprete de comandos de Windows el comando `usbipd wsl list`. El anterior comando listará los dispositivos conectados a la máquina.

Tras identificar cuál es el USB que se quiere utilizar desde WSL, ejecutar el comando de Windows

`usbipd wsl attach --busid <busid>` el cual asociará a WSL el dispositivo.



```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> usbipd wsl list
BUSID  VID:PID  DEVICE                                STATE
-----  -
1-1    0930:6544  Dispositivo de almacenamiento USB     Not attached
1-4    046d:c534  Dispositivo de entrada USB           Not attached
1-7    04f2:b5e0  VGA WebCam                            Not attached
1-10   04ca:3015  Bluetooth USB Module                 Not attached

PS C:\WINDOWS\system32> usbipd wsl attach --busid 1-1
usbipd: info: Using default WSL distribution 'Ubuntu'; specify the '--distribution' option to select a different one.
PS C:\WINDOWS\system32>
```

Figura 6.1.1: Listar y asociar dispositivo USB a WSL

Tras ello, en el explorador de archivos de Windows desaparecerá la conexión del dispositivo y pasará a visualizarse desde WSL con el comando `lsusb`. Además, se podrá interactuar con él mediante herramienta propias de Linux.

```
carlos@LAPTOP-BEA9J9V1: ~  
carlos@LAPTOP-BEA9J9V1:~$ lsusb  
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 001 Device 002: ID 0930:6544 Toshiba Corp. TransMemory-Mini / Kingston DataTraveler 2.0 Stick  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Figura 6.1.2: Dispositivo USB conectado a WSL

Sin embargo, tal y como se comenta en la introducción de este capítulo, todavía no es posible únicamente realizando estos pasos previos acceder al contenido del USB desde WSL sin hacer configuraciones adicionales. Por ello, una posibilidad es construir un *kernel* personalizado y habilitar el soporte para dichos dispositivos, tal y como se explica en este vídeo de YouTube del usuario *BonGuides*: [Modificación del kernel](#)

Una vez que se quiera dejar de usar el dispositivo en WSL, se puede desconectar directamente de manera física o ejecutar el siguiente comando de PowerShell: `usbipd wsl detach --busid`

Tras ello, en WSL se dejará de visualizar:

```
carlos@LAPTOP-BEA9J9V1: ~  
carlos@LAPTOP-BEA9J9V1:~$ lsusb  
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 001 Device 002: ID 0930:6544 Toshiba Corp. TransMemory-Mini / Kingston DataTraveler 2.0 Stick  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
carlos@LAPTOP-BEA9J9V1:~$ lsusb  
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Figura 6.1.3: Dispositivo USB desconectado de WSL

## 6.2. Acceso desde Windows a un disco externo formateado en ext4

Los pasos a realizar son los siguientes:

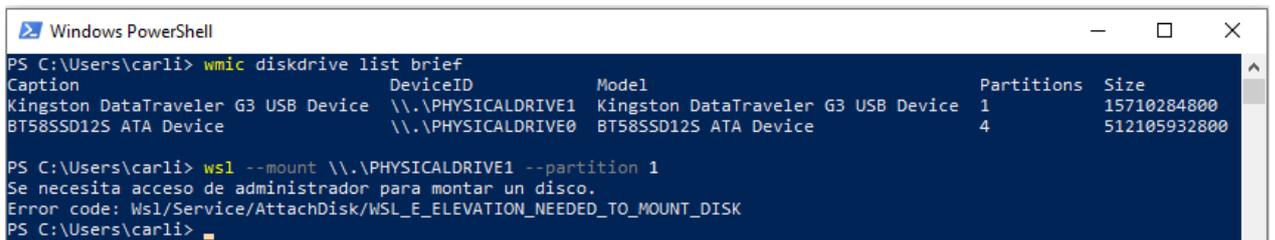
1. Ejecutar el intérprete de comando de Windows como *Administrador*
2. Averiguar el path del disco con el comando `wmic diskdrive list brief` (columna *device id*. También comprobar el valor de la columna *Partitions*).
3. Montar el disco en WSL mediante el comando `wsl --mount [PATH] --partition [PARTITION_NUMBER]`

```
Windows PowerShell  
PS C:\Users\carli> wmic diskdrive list brief  
Caption DeviceID Model Partitions Size  
Kingston DataTraveler G3 USB Device \\.\PHYSICALDRIVE1 Kingston DataTraveler G3 USB Device 1 15710284800  
BT58SSD12S ATA Device \\.\PHYSICALDRIVE0 BT58SSD12S ATA Device 4 512105932800  
PS C:\Users\carli>
```

Figura 6.2.1: Listado de unidades de disco conectadas a Windows

A veces puede ser necesario poner un número diferente al proporcionado por el comando en la columna *Partitions* correspondiente a la unidad en cuestión, por ejemplo, `--partition 2` cuando se muestra que solo hay 1 partición.

Cabe destacar que este método es solamente válido para montar discos HDD, no es una solución válida para memorias USB. Si se quisiera montar una unidad USB, aparecería el siguiente error:



```
Windows PowerShell
PS C:\Users\carli> wmic diskdrive list brief
Caption                                DeviceID                                Model                                    Partitions  Size
-----                                -
Kingston DataTraveler G3 USB Device    \\.\PHYSICALDRIVE1                    Kingston DataTraveler G3 USB Device    1           15710284800
BT58SSD12S ATA Device                 \\.\PHYSICALDRIVE0                    BT58SSD12S ATA Device                 4           512105932800

PS C:\Users\carli> wsl --mount \\.\PHYSICALDRIVE1 --partition 1
Se necesita acceso de administrador para montar un disco.
Error code: Wsl/Service/AttachDisk/WSL_E_ELEVATION_NEEDED_TO_MOUNT_DISK
PS C:\Users\carli>
```

*Figura 6.2.2: Mensaje de error al montar una unidad USB*

4. A partir de aquí, ya será visible el disco dentro de wsl en la ruta /mnt/wsl
5. Cuando se quiera dejar de utilizar el dispositivo desde WSL, hay que desmontarlo haciendo uso del comando `wsl --unmount [PATH]`

## 7. Ejecución de aplicaciones de GUI para Linux

Aunque es posible acceder al sistema de ficheros de WSL y ver/escribir/ejecutar cualquiera de ellos con el programa indicado instalado en Windows, WSL también admite la ejecución de aplicaciones de GUI de Linux en Windows con una experiencia de escritorio totalmente integrada. Por ello, es posible:

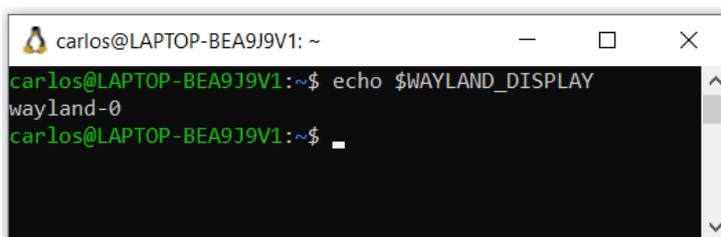
- Iniciar aplicaciones de Linux desde el menú Inicio de Windows
- Anclar aplicaciones de Linux a la barra de tareas de Windows
- Usar alt+tab para cambiar entre aplicaciones Linux y Windows
- Cortar y pegar entre aplicaciones de Windows y Linux

Al mismo tiempo, es necesario cumplir con los dos siguientes requisitos:

- Versión de Windows 10 o posteriores
- Contar con un controlador instalado para vGPU (*Virtual Graphics Processing Unit*).

### 7.1. Lanzar aplicaciones gráficas de WSL

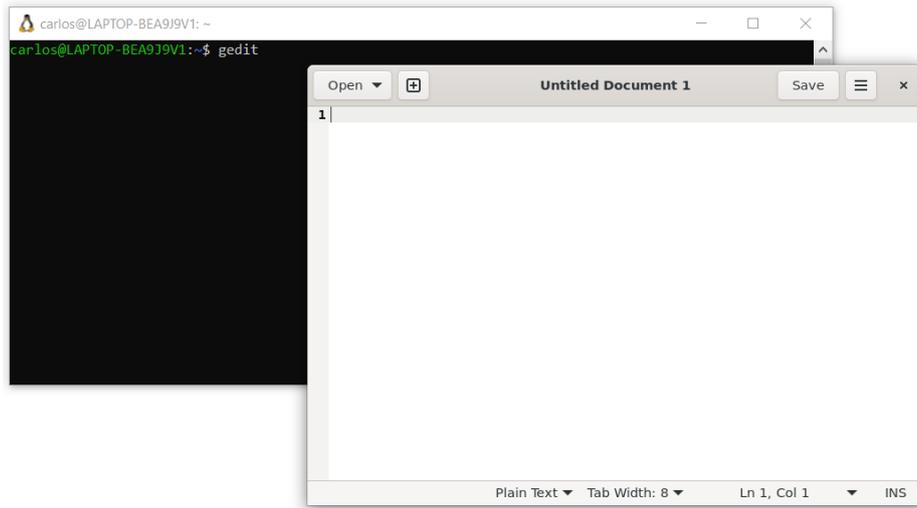
WSL usa por defecto *Wayland*, un servidor gráfico instalado que redirige directamente la salida gráfica de las aplicaciones a Windows.

A screenshot of a terminal window titled 'carlos@LAPTOP-BEA9J9V1: ~'. The terminal shows the command 'echo \$WAYLAND\_DISPLAY' being executed, with the output 'wayland-0' displayed on the next line. The prompt 'carlos@LAPTOP-BEA9J9V1:~\$' is visible at the end of the line.

```
carlos@LAPTOP-BEA9J9V1:~$ echo $WAYLAND_DISPLAY
wayland-0
carlos@LAPTOP-BEA9J9V1:~$
```

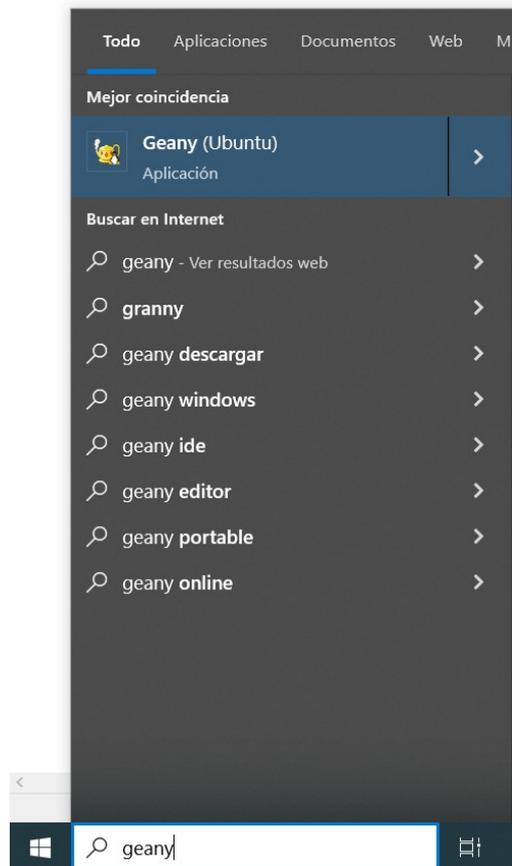
Figura 7.1.1: Servidor gráfico de WSL: Wayland

Por ello, es posible lanzar aplicaciones gráficas que hayan sido instaladas en WSL aprovechándose del entorno gráfico de Windows.



*Figura 7.1.2: Lanzar aplicación gráfica desde WSL*

Al mismo tiempo, también será posible lanzar dichas aplicaciones desde el explorador de Windows:



*Figura 7.1.3: Lanzar aplicación gráfica de WSL desde Windows*

Cabe destacar que hay aplicaciones diseñadas principalmente para ejecutarse en entornos Linux por lo que **su compatibilidad con WSL puede ser limitada**, es decir, ciertas aplicaciones que se instalen puede que no funcionen correctamente en WSL ya

que se está trabajando sobre Windows y, por ello, existen ciertas limitaciones con respecto a la compatibilidad de la aplicación. Esto se debe a que existen algunas diferencias y limitaciones en la forma en que se manejan los procesos y los recursos del sistema por lo que, dependiendo de la aplicación que se vaya a utilizar, puede ser necesario hacer una serie de configuraciones adicionales.

## 7.2. Lanzar archivos de vídeo o audio desde WSL

Al igual que con el servidor gráfico, WSL cuenta por defecto con un servidor de audio instalado que redirige su salida a Windows. El servidor que se está usando es *PulseAudio*.

Sin embargo, para poder ejecutar archivos de audio/vídeo es necesario contar con una aplicación que permita trabajar con estos tipos de archivos. Por ejemplo, el reproductor *VLC*.

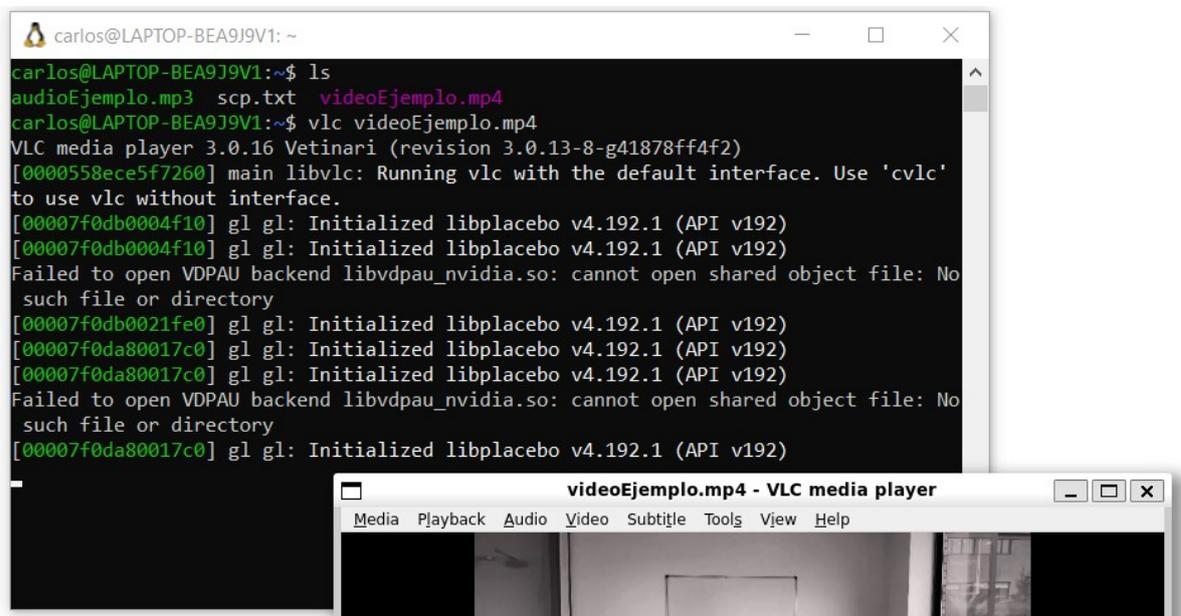


Figura 7.2.1: Redirección de audio y vídeo de WSL a Windows

## 8. Comparativa de rendimiento entre WSL y Linux Nativo

En este capítulo se realizará un análisis técnico para analizar el uso de WSL y Linux nativo. Para ello, se abordarán diversas comparativas relacionadas con el uso de la CPU, pruebas de red y estructuras de almacenamiento visibles desde cada entorno para comprender mejor cómo influye la virtualización y el rendimiento subyacente en la experiencia de estas dos alternativas. Primero se realizarán las mediciones en WSL y luego en Linux nativo para comparar los resultados.

Todas estas pruebas han sido realizadas en una máquina con arranque dual (Windows y Ubuntu). Aunque es una máquina un poco antigua, es completamente válida como ejemplo a estos efectos. Sus principales características son las siguientes:

- Marca y modelo: LENOVO-2324EZ5 (Lenovo ThinkPad X230).
- CPU: Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz, 2901 Mhz, 2 procesadores principales, 4 procesadores lógicos.
- Disco: KINGSTON SA400S37480G ATA Device, 480GB SSD
- RAM: 12,0GB

### 8.1. Benchmarks de CPU

Los benchmarks de CPU miden y evalúan el rendimiento y la capacidad de procesamiento de una unidad central de procesamiento (CPU) en un sistema informático.

Windows ofrece la posibilidad de cuatro modos de energía: *Ahorro de energía*, *Mejor batería*, *Mejor rendimiento*, *Máximo rendimiento*. Dado que las variaciones observadas en la realización de estos benchmarks al alternar entre estos modos eran del orden del 1%, posiblemente atribuibles al azar, solamente se harán las pruebas en único modo, el de *Mejor rendimiento*.

#### 1. sysbench

Para llevar a cabo esta sección, se hará uso de la herramienta *sysbench*. Esta herramienta de benchmarking de código abierto fue desarrollada por Alexey Kopytov y se utiliza para medir el rendimiento de la CPU, la velocidad de almacenamiento y otros componentes esenciales.

Comando: `sysbench cpu --time=10 run`

Al ejecutar el anterior comando, se establece una prueba de 10 segundos para medir el rendimiento de la CPU durante una prueba de benchmark. Dicho comando mostrará la siguiente información:

- Información de la versión de Sysbench y configuración de la prueba.

- Versión de Sysbench utilizada en la prueba.
- Opciones de configuración, como el tiempo de ejecución (en este caso, 10 segundos) y posibles ajustes específicos.
- Resultados de la prueba de rendimiento de la CPU.
  - Número de eventos (operaciones) que la CPU pudo realizar por segundo (eventos por segundo).
  - Estadísticas generales, como el tiempo total de ejecución y el número total de eventos realizados durante la prueba.
- Mediciones de latencia (tiempo de respuesta).
  - Valores de latencia mínima, promedio y máxima, que indican el tiempo que tomó a la CPU procesar operaciones en milisegundos.
  - Percentiles de latencia (Percentil 95), que indica que, en el 95% de los casos, la latencia está por debajo del valor indicado.
- Equidad de hilos (*threads fairness*).
  - Estadísticas sobre cómo se distribuyeron los eventos y el tiempo de ejecución entre los hilos utilizados en la prueba.

```

carloslenovo@DESKTOP-VJJU969: ~
carloslenovo@DESKTOP-VJJU969:~$ sysbench cpu --time=10 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 1107.34

General statistics:
  total time:                10.0008s
  total number of events:    11076

Latency (ms):
  min:                       0.87
  avg:                       0.90
  max:                       3.18
  95th percentile:          0.95
  sum:                       9996.07

Threads fairness:
  events (avg/stddev):       11076.0000/0.00
  execution time (avg/stddev): 9.9961/0.00

carloslenovo@DESKTOP-VJJU969:~$

```

Figura 8.1.1: Benchmark de CPU I en WSL

```

carlos@carlos-ThinkPad-X230: ~
carlos@carlos-ThinkPad-X230:~$ sysbench cpu --time=10 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 10000

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 1116.44

General statistics:
  total time:                10.0001s
  total number of events:    11167

Latency (ms):
  min:                        0.87
  avg:                        0.90
  max:                        2.77
  95th percentile:          0.99
  sum:                        9996.71

Threads fairness:
  events (avg/stddev):       11167.0000/0.00
  execution time (avg/stddev): 9.9967/0.00
carlos@carlos-ThinkPad-X230:~$

```

Figura 8.1.2: Benchmark de CPU I en Linux nativo

A continuación, se muestra una tabla y gráfica comparativa que resume los valores obtenidos durante la prueba:

Métrica	Linux Nativo	WSL
Velocidad de CPU	1116.44 eventos/s	1107.34 eventos/s
Latencia Mínima	0.87 ms	0.87 ms
Latencia Promedio	0.90 ms	0.90 ms
Latencia Máxima	2.77 ms	3.18 ms
Percentil 95	0.99 ms	0.95 ms
Suma de Latencias	9996.71 ms	9996.07 ms

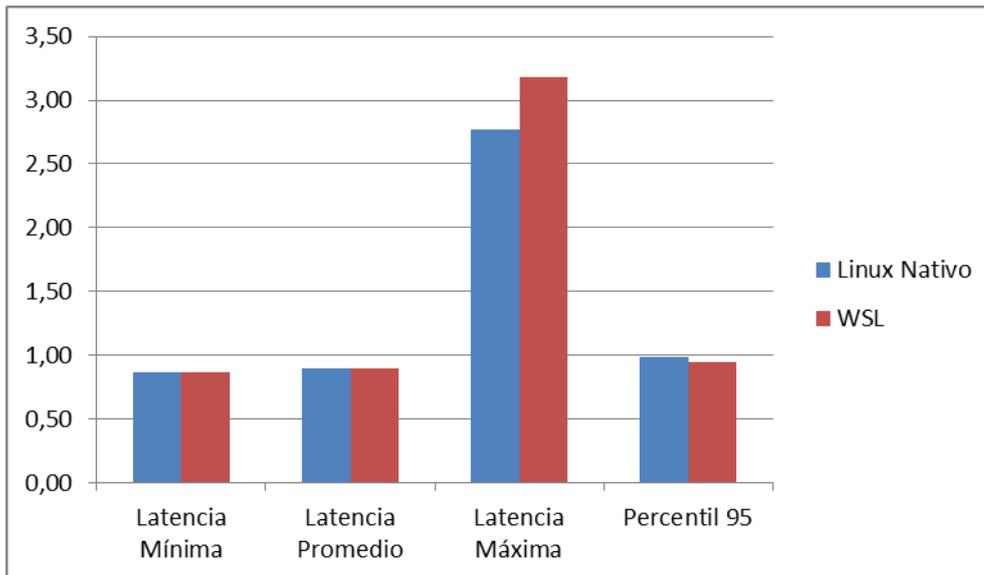


Figura 8.1.3: Gráfica de Benchmark de CPU I (1)

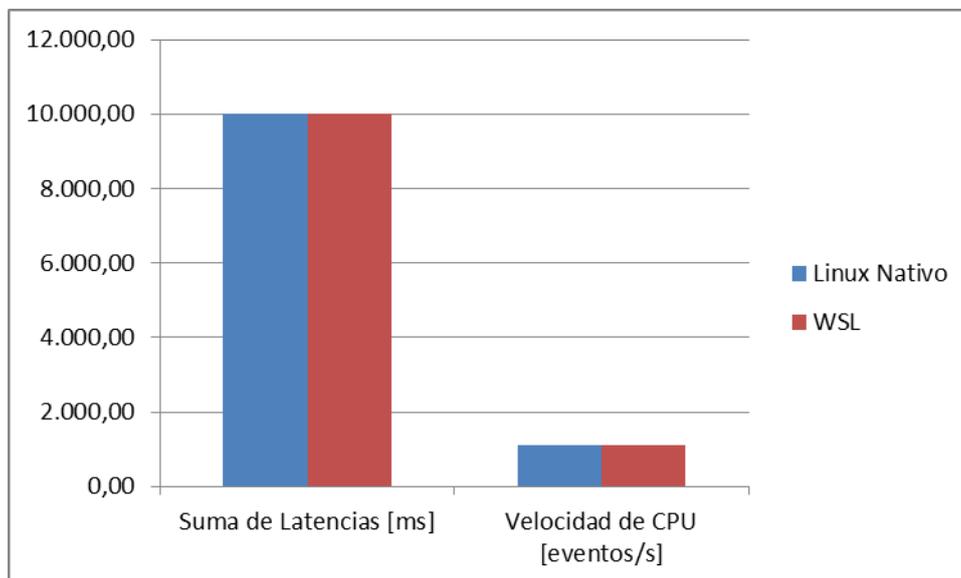


Figura 8.1.4: Gráfica de Benchmark de CPU I (2)

Las principales diferencias entre las pruebas de rendimiento de CPU I mostradas en las anteriores imágenes son las siguientes:

1. **Velocidad de CPU:** En el entorno de Linux nativo, se obtuvo una velocidad de CPU marginalmente mayor en términos de *eventos por segundo* (1116.44) en comparación con WSL (1107.34).
2. **Latencia:** Si bien las latencias mínima y promedio son similares entre ambos entornos, la latencia máxima en Linux nativo (2.77 ms) es más baja que en WSL (3.18 ms). Además, el percentil 95 en Linux nativo es más bajo (0.99 ms) en comparación con WSL (0.95 ms), lo que indica una latencia ligeramente mejor en el entorno nativo.

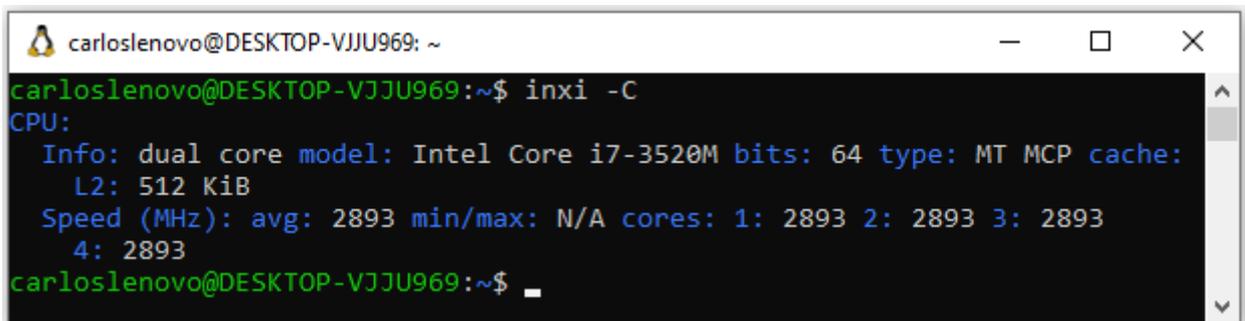
3. **Suma de Latencias:** La suma total de latencias en ambas pruebas es cercana (9996.71 ms en Linux nativo y 9996.07 ms en WSL). Esto sugiere que, en términos de la acumulación de tiempo de latencia, no hay diferencias significativas entre los dos entornos.
4. **Total de Eventos:** El número total de eventos registrados durante la prueba es marginalmente mejor en el entorno de Linux nativo (11167 eventos) en comparación con WSL (11076 eventos).
5. **Tiempo de Ejecución:** El tiempo de ejecución promedio es ligeramente menor en Linux nativo (9.9967 s) en comparación con WSL (9.9961 s).

Estas diferencias podrían atribuirse a factores como la virtualización y la capa de compatibilidad en WSL, que pueden tener un impacto en la medición precisa del rendimiento de la CPU. Sin embargo, las mediciones en ambos entornos son muy similares, casi iguales, por lo que no debería haber ninguna diferencia significativa.

## 2. inxi

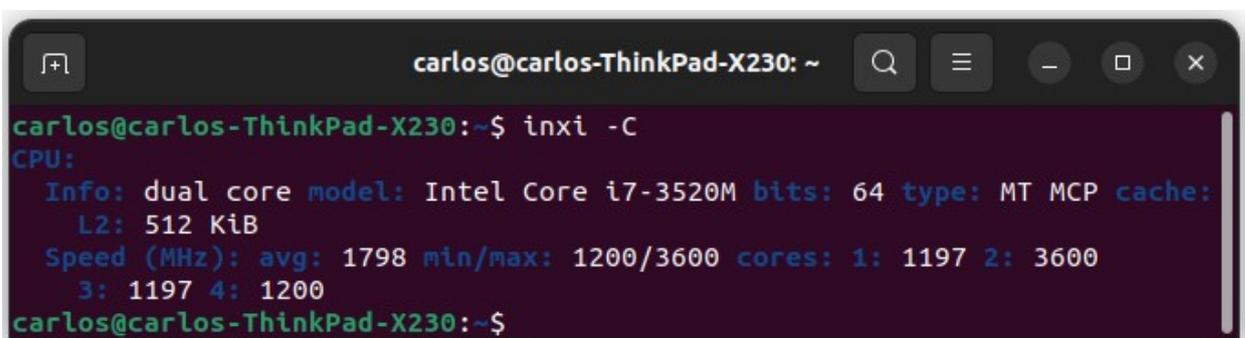
Para llevar a cabo esta sección, se hará uso de la herramienta *inxi*. Esta herramienta proporciona una descripción detallada de la CPU y otros componentes clave del sistema. A través de *inxi*, es posible acceder a información específica sobre el modelo de la CPU, la cantidad de núcleos y subprocesos, la velocidad base y turbo, entre otros detalles a tener en cuenta.

Comando: `inxi -C`



```
carloslenovo@DESKTOP-VJJU969: ~  
carloslenovo@DESKTOP-VJJU969:~$ inxi -C  
CPU:  
Info: dual core model: Intel Core i7-3520M bits: 64 type: MT MCP cache:  
L2: 512 KiB  
Speed (MHz): avg: 2893 min/max: N/A cores: 1: 2893 2: 2893 3: 2893  
4: 2893  
carloslenovo@DESKTOP-VJJU969:~$
```

Figura 8.1.5: Benchmark de CPU II en WSL



```
carlos@carlos-ThinkPad-X230: ~  
carlos@carlos-ThinkPad-X230:~$ inxi -C  
CPU:  
Info: dual core model: Intel Core i7-3520M bits: 64 type: MT MCP cache:  
L2: 512 KiB  
Speed (MHz): avg: 1798 min/max: 1200/3600 cores: 1: 1197 2: 3600  
3: 1197 4: 1200  
carlos@carlos-ThinkPad-X230:~$
```

Figura 8.1.6: Benchmark de CPU II en Linux nativo

A continuación, se muestra una tabla y gráfica comparativa que resume los valores obtenidos durante la prueba:

Diferencias	Linux Nativo	WSL
Velocidad de la CPU Promedio [MHz]	1798	2893
Rango de Velocidad Mínima [MHz]	1200	N/A para mínimo
Rango de Velocidad Máxima [MHz]	3600	N/A para máximo
Velocidad núcleo 1 [MHz]	1197	2893
Velocidad núcleo 2 [MHz]	3600	2893
Velocidad núcleo 3 [MHz]	1197	2893
Velocidad núcleo 4 [MHz]	1200	2893

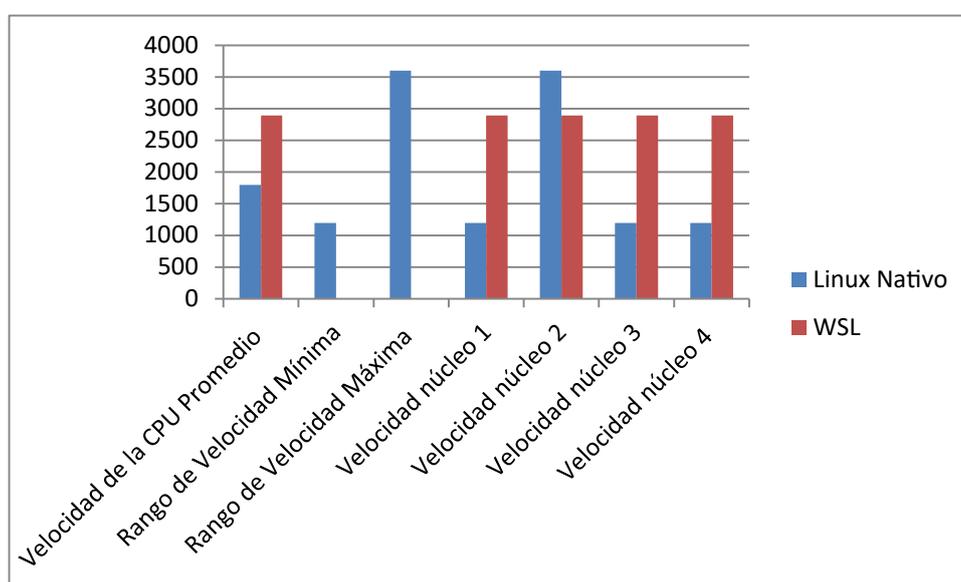


Figura 8.1.7: Gráfica de Benchmark de CPU II

Las principales diferencias observadas en la salida del comando `inxi -C` entre Linux nativo y WSL son las siguientes:

1. **Velocidad de la CPU Promedio:** En el entorno nativo, la velocidad promedio de la CPU es de aproximadamente 1798 MHz, mientras que en WSL es de alrededor de 2893 MHz. Esto sugiere que la CPU puede operar a una velocidad más alta en WSL.
2. **Rango de Velocidad Mínima/Maxima:** En el entorno nativo, la CPU tiene un rango de velocidad mínimo/máximo de 1200/3600 MHz, mientras que en WSL, solo se muestra el valor promedio (N/A para mínimo/máximo). Esto podría indicar que la detección de los valores mínimo y máximo de velocidad podría ser diferente en WSL.
3. **Velocidades Individuales de los Núcleos:** En el entorno nativo, se muestra una variedad de velocidades entre los núcleos, lo que indica diferencias de velocidad entre los núcleos. En WSL, se muestra que todos los núcleos operan a la misma velocidad.

Las diferencias en las velocidades promedio y los valores mínimo/máximo de la CPU sugieren una posible variación en la detección de la velocidad de la CPU entre Linux nativo y WSL. Además, las diferencias en las velocidades individuales de los núcleos podrían indicar una administración de energía o ajustes de rendimiento diferentes entre los dos entornos. Estas diferencias podrían deberse a las particularidades de cada entorno y su relación con el hardware subyacente.

### 3. Número de BogoMIPS

Comando: `dmesg | grep -i bogo`

Al ejecutar el anterior comando, se obtiene información sobre el número de BogoMIPS, que es una medida aproximada propia de Linux de los MIPS (Millones de Instrucciones por Segundo), que sirve para medir el rendimiento de la CPU y que se utiliza en sistemas basados en la arquitectura x86.

```

carloslenovo@DESKTOP-VJUU969: ~
carloslenovo@DESKTOP-VJUU969:~$ dmesg |grep -i bogo
[   0.838616] Calibrating delay loop (skipped), value calculated using
timer frequency.. 5786.83 BogoMIPS (lpj=28934150)
[   0.970984] smpboot: Total of 4 processors activated (23147.32 Bog
MIPS)
carloslenovo@DESKTOP-VJUU969:~$

```

Figura 8.1.8: Benchmark de CPU III en WSL

```

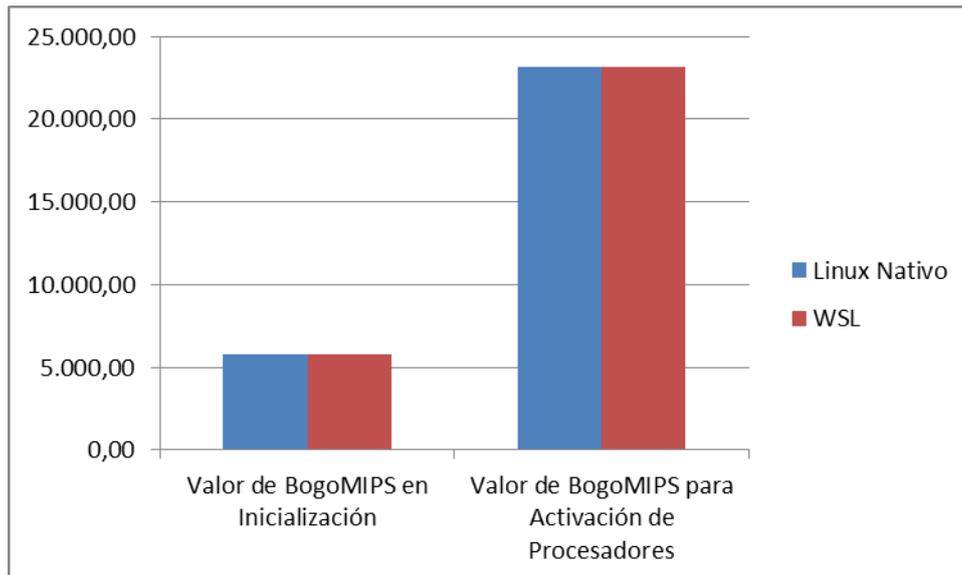
carlos@carlos-ThinkPad-X230: ~
carlos@carlos-ThinkPad-X230:~$ sudo dmesg | grep -i bogo
[   0.111566] Calibrating delay loop (skipped), value calculated using timer
frequency.. 5786.98 BogoMIPS (lpj=11573972)
[   0.146040] smpboot: Total of 4 processors activated (23147.94 BogoMIPS)
carlos@carlos-ThinkPad-X230:~$

```

Figura 8.1.9: Benchmark de CPU III en Linux nativo

A continuación, se muestra una tabla y gráfica comparativa que resume los valores obtenidos durante la prueba:

Características	Linux Nativo	WSL
Valor de BogoMIPS en Inicialización	5786.98	5786.83
Valor de BogoMIPS para Activación de Procesadores	23147.9	23147.32



*Figura 8.1.10: Gráfica de Benchmark de CPU III*

Las principales diferencias entre las pruebas de rendimiento de CPU III mostradas en las anteriores imágenes son las siguientes:

1. **Valor de Bogomips en la inicialización:** En el entorno de Linux nativo, el valor de Bogomips calculado durante la inicialización es prácticamente idéntico, con un valor de 5786.98 en comparación con 5786.83 en WSL.
2. **Valor de Bogomips para la Activación de Procesadores:** De forma similar, el valor de Bogomips registrado durante la activación de procesadores es prácticamente idéntico en ambos casos.
3. **Precisión del Valor de Bogomips:** Las cifras decimales en los valores de Bogomips son diferentes entre los dos entornos. Esto podría deberse a pequeñas variaciones en la forma en que se calcula y registra el valor durante el proceso de inicialización.

Aunque las diferencias son mínimas, se observa una ligera variación en los valores de Bogomips entre el entorno de Linux nativo y WSL. Estas diferencias podrían ser atribuidas a la configuración del sistema, el proceso de inicialización y la virtualización utilizada en WSL, lo que puede influir en la precisión de los valores registrados.

**Conclusión general de las pruebas realizadas:** aunque algunos de los parámetros mostrados en las salidas de los anteriores comandos han variado dependiendo del entorno donde se han ejecutado, las diferencias observadas son sutiles y, en la mayoría de los casos, es poco probable que tengan un impacto significativo en el uso cotidiano del sistema.

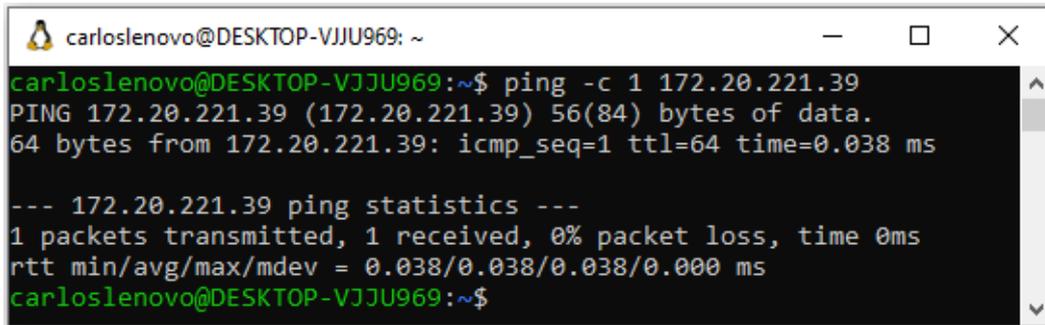
En algún caso, puede ser posible que las mediciones de CPU resulten ser bastante mejores en WSL. Sin embargo, no se tiene una explicación para este fenómeno, tal vez sea un error debido a las herramientas de medición.

## 8.2. Comparación de interfaces de red y Benchmarks de red

Los benchmarks de red miden la tasa de transferencia máxima entre un cliente y un servidor (network throughput). También se puede traducir por ancho de banda digital, rendimiento de red o caudal de datos.

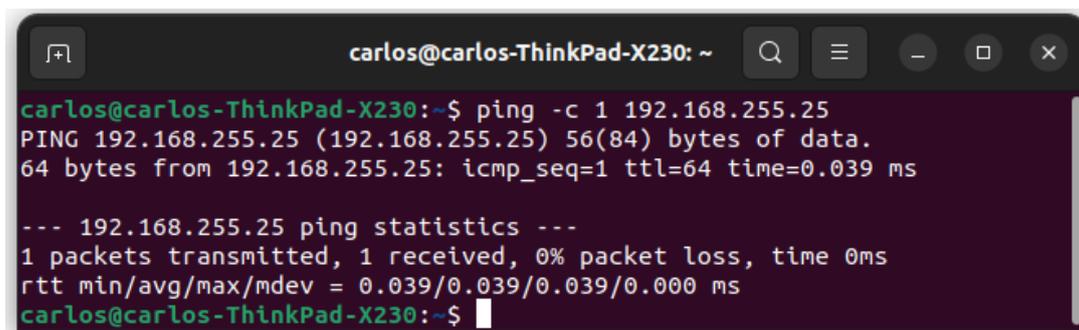
### 1. Ping a la máquina local

Comando: `ping -c 1 <ip_máquina>`



```
carloslenovo@DESKTOP-VJJU969: ~  
carloslenovo@DESKTOP-VJJU969:~$ ping -c 1 172.20.221.39  
PING 172.20.221.39 (172.20.221.39) 56(84) bytes of data.  
64 bytes from 172.20.221.39: icmp_seq=1 ttl=64 time=0.038 ms  
  
--- 172.20.221.39 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 0.038/0.038/0.038/0.000 ms  
carloslenovo@DESKTOP-VJJU969:~$
```

Figura 8.2.1: Ping en WSL



```
carlos@carlos-ThinkPad-X230: ~  
carlos@carlos-ThinkPad-X230:~$ ping -c 1 192.168.255.25  
PING 192.168.255.25 (192.168.255.25) 56(84) bytes of data.  
64 bytes from 192.168.255.25: icmp_seq=1 ttl=64 time=0.039 ms  
  
--- 192.168.255.25 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 0.039/0.039/0.039/0.000 ms  
carlos@carlos-ThinkPad-X230:~$
```

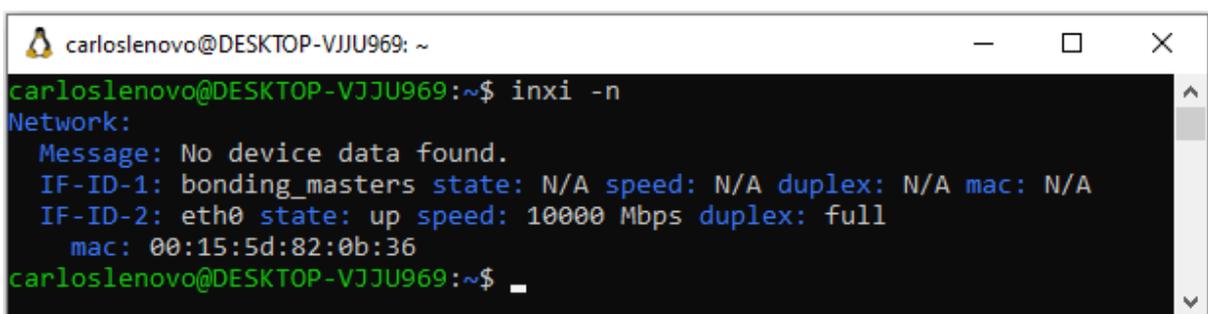
Figura 8.2.2: Ping en Linux nativo

En cuanto a la ejecución de un *ping*, no se observan diferencias significativas en el tiempo requerido para recibir el paquete entre WSL y el entorno de Linux nativo.

### 2. inxi

A continuación, se va a hacer uso de la herramienta *inxi* para obtener información general sobre las interfaces de red visibles desde cada entorno.

Comando: `inxi -n`



```
carloslenovo@DESKTOP-VJJU969: ~  
carloslenovo@DESKTOP-VJJU969:~$ inxi -n  
Network:  
Message: No device data found.  
IF-ID-1: bonding_masters state: N/A speed: N/A duplex: N/A mac: N/A  
IF-ID-2: eth0 state: up speed: 10000 Mbps duplex: full  
mac: 00:15:5d:82:0b:36  
carloslenovo@DESKTOP-VJJU969:~$
```

Figura 8.2.3: Interfaces de red visibles en WSL

```
carlos@carlos-ThinkPad-X230: ~  
carlos@carlos-ThinkPad-X230:~$ inxi -n  
Network:  
Device-1: Intel 82579LM Gigabit Network driver: e1000e  
IF: enp0s25 state: down mac: 3c:97:0e:b8:0a:29  
Device-2: Intel Centrino Advanced-N 6205 [Taylor Peak]  
driver: iwlwifi  
IF: wlp3s0 state: up mac: a4:4e:31:0c:be:2c  
IF-ID-1: wwan0 state: down mac: 36:63:4b:5d:72:b4  
carlos@carlos-ThinkPad-X230:~$
```

Figura 8.2.4: Interfaces de red visibles en Linux nativo

Las diferencias clave en la información de red obtenida mediante `inxi -n` entre Linux nativo y WSL son las siguientes:

#### Linux Nativo - Interfaces de Red:

- Se identifican tres interfaces de red: `enp0s25`, `wlp3s0` y `wwan0`.
- `enp0s25` (Ethernet) y `wlp3s0` (Wi-Fi) tienen un estado definido, con `enp0s25` en estado inactivo (`down`) y `wlp3s0` en estado activo (`up`).
- `wwan0` está inactivo (`down`).

#### WSL - Interfaces de Red:

- La salida informa que no se encontraron datos de dispositivo.
- Muestra dos interfaces, `bonding_masters` y `eth0`.
- `eth0` está en estado activo (`up`) con una velocidad de 10000 Mbps y dúplex completo.

Las diferencias sugieren que la detección de interfaces de red y su estado es más detallada y precisa en Linux nativo en comparación con WSL. Mientras que en Linux nativo se reconocen las interfaces `enp0s25`, `wlp3s0` y `wwan0`, en WSL se muestra información menos detallada y no se detecta información específica sobre las interfaces de red. Esto podría deberse a las diferencias en cómo WSL interactúa con el hardware de red en comparación con un sistema operativo Linux nativo.

### 3. netstat

Haciendo uso del comando `netstat`, se va a listar la información sobre las conexiones UDP y TCP activas y puertos abiertos.

```

carloslenovo@DESKTOP-VJJU969: ~
carloslenovo@DESKTOP-VJJU969:~$ netstat -tu
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
carloslenovo@DESKTOP-VJJU969:~$ netstat -lut
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 127.0.0.53:domain      0.0.0.0:*               LISTEN
udp      0      0 127.0.0.53:domain      0.0.0.0:*
udp      0      0 localhost:323           0.0.0.0:*
udp6     0      0 ip6-localhost:323     [::]:*
carloslenovo@DESKTOP-VJJU969:~$

```

Figura 8.2.5: Conexiones TCP y UDP visibles en WSL

```

carlos@carlos-ThinkPad-X230: ~
carlos@carlos-ThinkPad-X230:~$ netstat -tu
Conexiones activas de Internet (servidores w/o)
Proto  Recib Enviad Dirección local           Dirección remota          Estado
tcp     0      0 carlos-ThinkPad-X:38664  104.18.3.161:https       ESTABLECIDO
tcp     0      0 carlos-ThinkPad-X:34146  52.108.9.12:https        ESTABLECIDO
tcp     0      0 carlos-ThinkPad-X:36050  55.65.117.34.bc.g:https  ESTABLECIDO
tcp     0      0 carlos-ThinkPad-X:46698  52.111.236.11:https      ESTABLECIDO
tcp     0      0 carlos-ThinkPad-X:37370  ec2-52-44-101-55.:https  TIME_WAIT
tcp     0      0 carlos-ThinkPad-X:42794  52.108.240.24:https      ESTABLECIDO
tcp     0      0 carlos-ThinkPad-X:51512  181.214.120.34.bc:https  ESTABLECIDO
udp     0      0 carlos-ThinkPad-:bootpc  _gateway:bootps         ESTABLECIDO
carlos@carlos-ThinkPad-X230:~$ netstat -lut
Conexiones activas de Internet (solo servidores)
Proto  Recib Enviad Dirección local           Dirección remota          Estado
tcp     0      0 localhost:domain         0.0.0.0:*                ESCUCHAR
tcp     0      0 localhost:ipp            0.0.0.0:*                ESCUCHAR
tcp6    0      0 ip6-localhost:ipp       [::]:*                   ESCUCHAR
udp     0      0 localhost:domain         0.0.0.0:*                ESCUCHAR
udp     0      0 0.0.0.0:631             0.0.0.0:*                ESCUCHAR
udp     0      0 0.0.0.0:37246           0.0.0.0:*                ESCUCHAR
udp     0      0 0.0.0.0:mdns            0.0.0.0:*                ESCUCHAR
udp6    0      0 [::]:51080              [::]:*                   ESCUCHAR
udp6    0      0 [::]:mdns                [::]:*                   ESCUCHAR
carlos@carlos-ThinkPad-X230:~$

```

Figura 8.2.6: Conexiones TCP y UDP visibles en Linux nativo

Las diferencias más destacadas en las salidas de `netstat` entre Linux nativo y WSL son las siguientes:

Linux Nativo - Conexiones de Internet (servidores w/o)

- Se muestran conexiones TCP y UDP activas con direcciones locales y remotas, y su estado.
- Varias conexiones TCP están en estado *ESTABLECIDO*.
- También se muestra una conexión UDP a través de *bootpc* (Boot Protocol Client).

Linux Nativo - Conexiones de Internet (solo servidores):

- Se muestran puertos locales TCP y UDP que escuchan conexiones entrantes.

- Los puertos locales *domain* (DNS) e *ipp* (Internet Printing Protocol) están en estado *ESCUCHAR*.

#### WSL - Conexiones de Internet (w/o servers):

- No se muestran conexiones activas sin servidores en esta salida.

#### WSL - Conexiones de Internet (solo servidores):

- Se muestra un puerto local TCP (*domain*) y puertos locales UDP (*domain*, 323) que escuchan conexiones entrantes.

Las diferencias entre las salidas de `netstat` entre Linux nativo y WSL son notables. En Linux nativo, se muestran conexiones activas en detalle, incluyendo conexiones *ESTABLECIDAS*, mientras que en WSL, la salida es más limitada y muestra principalmente puertos locales que escuchan conexiones. Estas diferencias se deben a la forma en que WSL interactúa con el sistema de red subyacente y la capacidad para acceder a ciertos tipos de conexiones, que son gestionadas directamente por Windows. Es decir, WSL tiene menos servicios en ejecución que la máquina Linux nativa, pero ha de tenerse en cuenta que todos los servicios de la máquina Windows no son visibles desde aquí.

## 4. `iperf`

Haciendo uso de la herramienta `iperf`, se va a medir el rendimiento de red que proporciona Linux nativo y WSL. Para ello, se utilizará una máquina externa que actuará como servidor y que está conectada a la misma red. Esta máquina externa también está corriendo WSL. La máquina cliente será la máquina que tiene instalado Linux nativo y WSL.

Cabe destacar que WSL no cuenta por defecto con la posibilidad de actuar como servidor de `iperf` aunque sí como cliente. Para que WSL pueda actuar como servidor, es necesario redirigir el puerto sobre el que trabaja `iperf` a WSL realizando la misma configuración de la sección 4.1.1 a partir del paso cuatro teniendo en cuenta cuáles son los puertos que se quieren redirigir. En este caso de prueba, el puerto introducido fue el 5001.

Comando máquina servidor: `iperf -s`

Comando máquina cliente: `iperf -c <ip_servidor>`

### 4.1 Linux Nativo

```
carlos@LAPTOP-BEA9J9V1: ~
carlos@LAPTOP-BEA9J9V1:~$ iperf -s
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 172.31.228.108 port 5001 connected with 172.31.224.1 port 64269
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.0455 sec  446 MBytes  373 Mbits/sec
^Ccarlos@LAPTOP-BEA9J9V1:~$
```

Figura 8.2.7: Benchmark de red I: Servidor iperf en WSL (1)

```
carlosgp@carlosgp-ThinkPad-X230: ~
carlosgp@carlosgp-ThinkPad-X230:~$ iperf -c 192.168.1.24
-----
Client connecting to 192.168.1.24, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[ 1] local 192.168.1.29 port 46232 connected with 192.168.1.24 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.0631 sec  446 MBytes  372 Mbits/sec
carlosgp@carlosgp-ThinkPad-X230:~$ S
```

Figura 8.2.8: Benchmark de red I: Cliente iperf en Linux nativo

## 4.2 WSL

```
carlos@LAPTOP-BEA9J9V1: ~
carlos@LAPTOP-BEA9J9V1:~$ iperf -s
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 1] local 172.31.228.108 port 5001 connected with 172.31.224.1 port 64251
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.1913 sec  353 MBytes  291 Mbits/sec
^Ccarlos@LAPTOP-BEA9J9V1:~$
```

Figura 8.2.9: Benchmark de red I: Servidor iperf en WSL (2)

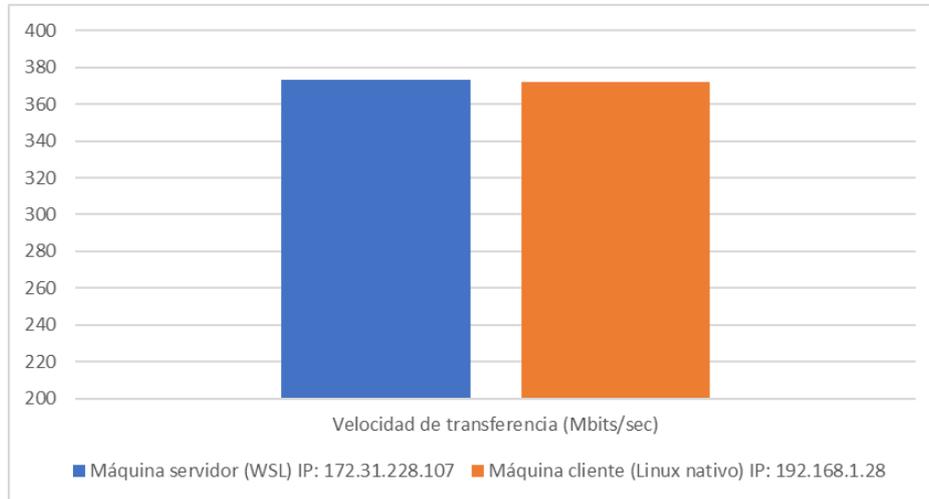
```
carlos@DESKTOP-6BKRT25: ~
carlos@DESKTOP-6BKRT25:~$ iperf -c 192.168.1.24
-----
Client connecting to 192.168.1.24, TCP port 5001
TCP window size: 85.0 KByte (default)
-----
[ 1] local 172.31.192.64 port 45768 connected with 192.168.1.24 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 1] 0.0000-10.2083 sec  353 MBytes  290 Mbits/sec
carlos@DESKTOP-6BKRT25:~$
```

Figura 8.2.10: Benchmark de red I: Cliente iperf en WSL

Los datos más importantes a destacar de ambas pruebas son los siguientes:

Primera medición, Servidor WSL y Cliente Linux nativo:

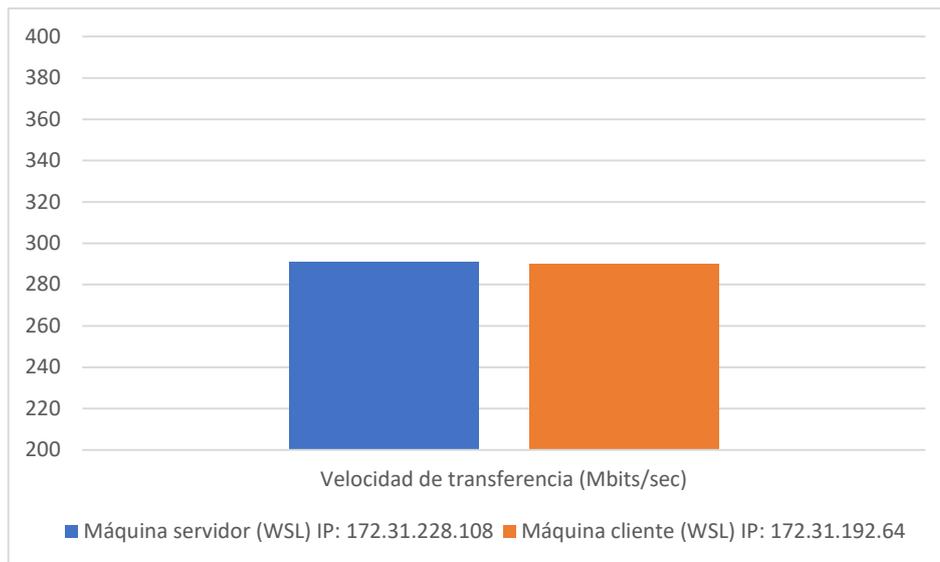
	<b>Máquina servidor (WSL)</b>	<b>Máquina cliente (Linux nativo)</b>
<b>Dirección IP</b>	172.31.228.108	192.168.1.29
<b>Velocidad de transferencia</b>	373 Mbts/sec	372 Mbts/sec



*Figura 8.2.11: Gráfica de Benchmark de red I: Cliente iperf en Linux nativo*

Segunda medición, Servidor WSL y Cliente WSL:

	<b>Máquina servidor (WSL)</b>	<b>Máquina cliente (WSL)</b>
<b>Dirección IP</b>	172.31.228.108	172.31.192.64
<b>Velocidad de transferencia</b>	291 Mbts/sec	290 Mbts/sec



*Figura 8.2.12: Gráfica de Benchmark de red I: Cliente iperf en WSL*

Basándose en estas mediciones, se puede concluir que en este caso específico, el rendimiento de la red en ambos casos ha sido prácticamente idéntico. Sin embargo, las diferencias existentes pueden atribuirse a factores externos tales como la carga de la red en el momento de realización de la prueba. Por ello, se realizaron varias pruebas en diferentes instantes de tiempo y la diferencia de valores variaba en valores comprendidos entre  $\pm 10\text{MB}$  con respecto a las mediciones actuales.

Por ello, en ambos casos los valores de velocidad de transferencia son más que aceptables para su correcto funcionamiento.

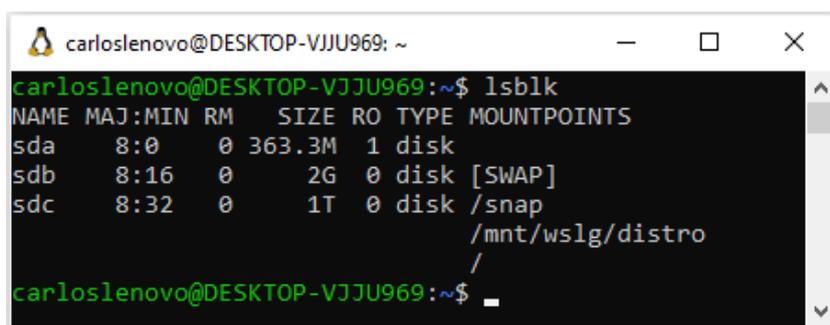
**Conclusión general de todas las pruebas realizadas:** si bien WSL ofrece un entorno Linux dentro de Windows, existen diferencias en la forma en que interactúa con la red en comparación con un sistema operativo Linux nativo. Estas diferencias pueden influir en la conectividad y la administración de red entre los dos entornos. Esto se debe principalmente a que la configuración de red en WSL se gestiona a través de Windows, por lo que no se puede cambiar la configuración de red directamente desde WSL.

### 8.3. Comparación de estructuras de almacenamiento y Benchmarks de operaciones de lectura y escritura

En esta sección se llevará a cabo un análisis comparativo de la información que muestra la herramienta `lsblk` y la herramienta `hdparm` al ser utilizada en los entornos de Linux nativo y WSL.

#### 1. `lsblk`

Esta herramienta se utiliza para listar información sobre dispositivos de bloque, como discos duros, unidades SSD y particiones. Proporciona una vista estructurada y fácil de entender de la estructura de almacenamiento en tu sistema.



```
carloslenovo@DESKTOP-VJJU969: ~  
carloslenovo@DESKTOP-VJJU969:~$ lsblk  
NAME MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS  
sda   8:0    0 363.3M  1 disk  
sdb   8:16   0     2G  0 disk [SWAP]  
sdc   8:32   0     1T  0 disk /snap  
                               /mnt/wslg/distro  
                               /  
carloslenovo@DESKTOP-VJJU969:~$
```

Figura 8.3.1: Comparación de dispositivos de bloque I en WSL

```
carlos@carlos-ThinkPad-X230: ~  
carlos@carlos-ThinkPad-X230:~$ lsblk  
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS  
loop0 7:0 0 63,3M 1 loop /snap/core20/1822  
loop1 7:1 0 4K 1 loop /snap/bare/5  
loop2 7:2 0 240,6M 1 loop /snap/firefox/2356  
loop3 7:3 0 346,3M 1 loop /snap/gnome-3-38-2004/119  
loop4 7:4 0 91,7M 1 loop /snap/gtk-common-themes/1535  
loop5 7:5 0 45,9M 1 loop /snap/snap-store/638  
loop6 7:6 0 49,8M 1 loop /snap/snapd/18357  
loop7 7:7 0 304K 1 loop /snap/snapd-desktop-integration/49  
sda 8:0 0 476,9G 0 disk  
├─sda1 8:1 0 50M 0 part  
├─sda2 8:2 0 263,1G 0 part  
├─sda3 8:3 0 1K 0 part  
├─sda4 8:4 0 186,3G 0 part /var/snap/firefox/common/host-hunspell  
└─sda5 8:5 0 27G 0 part /boot/efi  
carlos@carlos-ThinkPad-X230:~$
```

Figura 8.3.2: Comparación de dispositivos de bloque I en Linux nativo

Como se puede observar en las imágenes, la salida del comando `lsblk` en WSL y Linux nativo es diferente porque WSL y Linux nativo tienen diferentes conjuntos de dispositivos de bloque disponibles. En Linux nativo, `lsblk` muestra información sobre todos los dispositivos de bloque disponibles en el sistema, incluidos discos duros, unidades USB, particiones, etc. En WSL, `lsblk` solo muestra información sobre los dispositivos de bloque virtuales proporcionados por WSL, que pueden ser diferentes de los dispositivos de bloque disponibles en el sistema host de Windows.

## 2. Benchmark de escritura y lectura

### 2.1 Benchmark de tiempo de escritura

Comando: `time dd if=/dev/zero of=testfile bs=1M count=1000`

Este emplea dos herramientas principales: `dd` y `time`. La primera se utiliza para crear un archivo llamado `testfile` lleno de ceros en la ruta `/dev/zero`, con parámetros que especifican el tamaño de bloque y la cantidad de bloques. La segunda, `time`, se utiliza para medir el tiempo que lleva ejecutar la operación `dd`, proporcionando información sobre la duración de la operación al final de la ejecución del comando. Este comando crea un archivo de tamaño específico relleno con ceros y mide el tiempo necesario para realizar esta tarea.

```
carlos@DESKTOP-6BKRT25: ~  
carlos@DESKTOP-6BKRT25:~$ time dd if=/dev/zero of=testfile bs=1M count=1000  
1000+0 records in  
1000+0 records out  
1048576000 bytes (1.0 GB, 1000 MiB) copied, 0.795389 s, 1.3 GB/s  
  
real 0m0.803s  
user 0m0.000s  
sys 0m0.793s  
carlos@DESKTOP-6BKRT25:~$
```

Figura 8.3.3: Benchmark de tiempo de escritura en WSL

```

carlosgp@carlosgp-ThinkPad-X230: ~
carlosgp@carlosgp-ThinkPad-X230:~$ time dd if=/dev/zero of=testfile bs=1M count=1000
1000+0 registros leídos
1000+0 registros escritos
1048576000 bytes (1,0 GB, 1000 MiB) copied, 0,831844 s, 1,3 GB/s

real    0m0,835s
user    0m0,004s
sys     0m0,819s
carlosgp@carlosgp-ThinkPad-X230:~$

```

Figura 8.3.4: Benchmark de tiempo de escritura en Linux nativo

A continuación, se muestran los datos más relevantes resumidos en una tabla:

	Tiempo Real (real)	Tiempo de Usuario (user)	Tiempo del Sistema (sys)	Velocidad de Escritura (GB/s)
<b>WSL</b>	0.803 segundos	0.000 segundos	0.793 segundos	1.3 GB/s
<b>Linux nativo</b>	0.835 segundos	0.004 segundos	0.819 segundos	1.3 GB/s

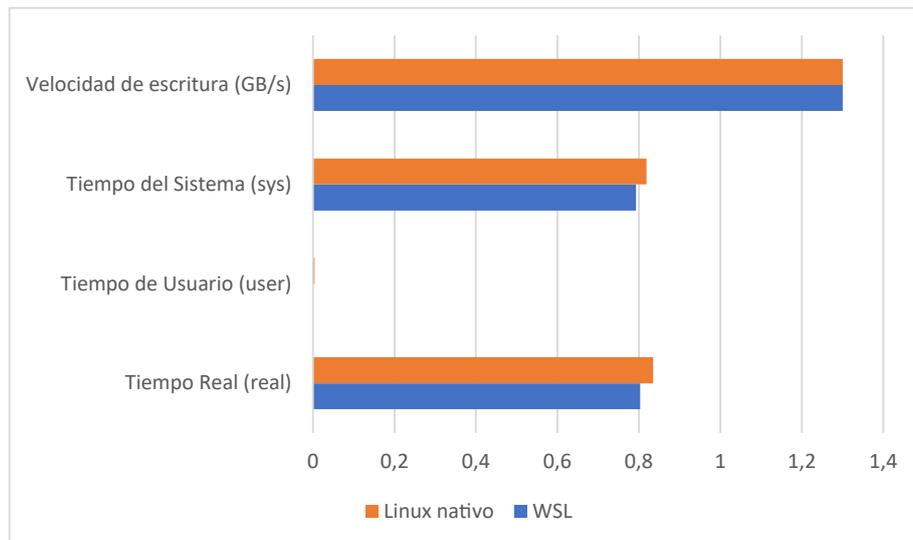


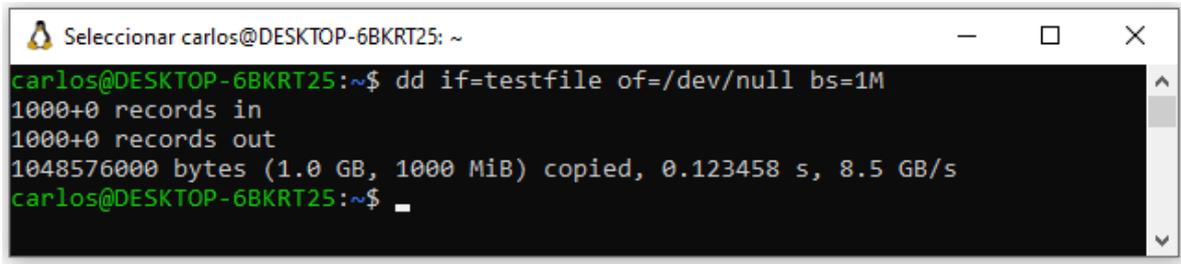
Figura 8.3.5: Gráfica de Benchmark de tiempo de escritura

La diferencia de tiempo entre WSL y Linux nativo es muy pequeña y podría atribuirse a variaciones menores en la carga del sistema y el rendimiento de E/S (entrada/salida). En ambos casos, la velocidad de escritura (copied) ha sido similar, alrededor de 1.3 GB/s, lo que indica que la operación de escritura en el archivo *testfile* se realizó con eficiencia en ambos entornos.

## 2.1 Benchmark de tiempo de lectura

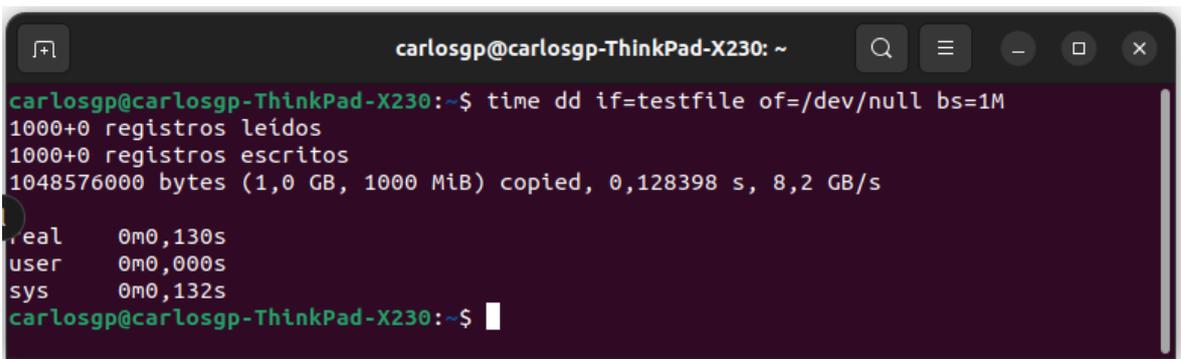
Comando: `time dd if=testfile of=/dev/null bs=1M`

Este comando usa las mismas herramientas explicadas en el anterior apartado y se utiliza para eliminar el contenido del fichero *testfile* midiendo el tiempo que lleva vaciar el archivo en */dev/null*.



```
Selecionar carlos@DESKTOP-6BKRT25: ~
carlos@DESKTOP-6BKRT25:~$ dd if=testfile of=/dev/null bs=1M
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 0.123458 s, 8.5 GB/s
carlos@DESKTOP-6BKRT25:~$
```

Figura 8.3.6: Benchmark de tiempo de lectura en WSL



```
carlosgp@carlosgp-ThinkPad-X230: ~
carlosgp@carlosgp-ThinkPad-X230:~$ time dd if=testfile of=/dev/null bs=1M
1000+0 registros leídos
1000+0 registros escritos
1048576000 bytes (1,0 GB, 1000 MiB) copied, 0,128398 s, 8,2 GB/s
real    0m0,130s
user    0m0,000s
sys     0m0,132s
carlosgp@carlosgp-ThinkPad-X230:~$
```

Figura 8.3.7: Benchmark de tiempo de lectura en Linux nativo

A continuación, se muestran los datos más relevantes resumidos en una tabla:

	<b>Tiempo Real (real)</b>	<b>Velocidad de Lectura (GB/s)</b>
WSL	0.123 segundos	8.5 GB/s
Linux	0.130 segundos	8.2 GB/s

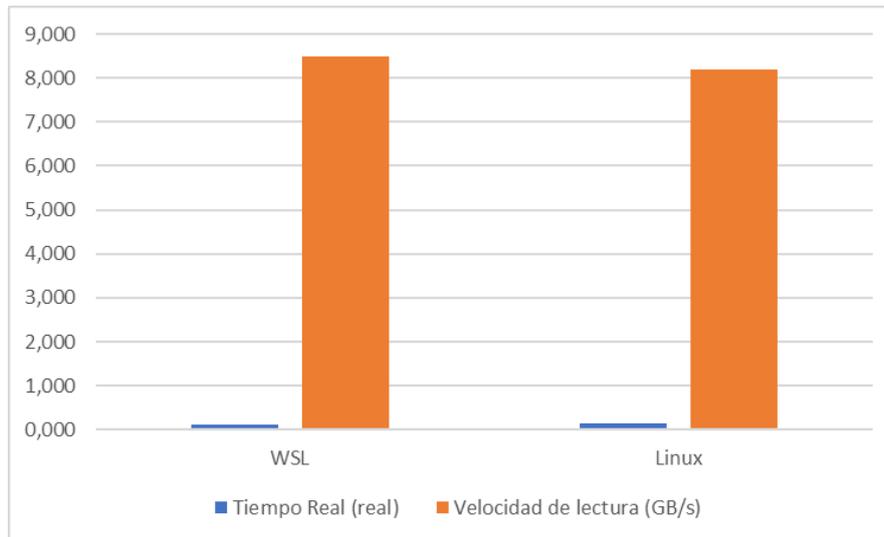


Figura 8.3.8: Gráfica de Benchmark de tiempo de lectura

Se observa que en ambos casos hay un rendimiento muy similar en términos de velocidad de lectura desde *testfile* hacia */dev/null*. Ambos sistemas alcanzan velocidades de lectura superiores a 8 GB/s, y la diferencia en el tiempo real entre ellos es mínima, con WSL siendo ligeramente más rápido en este caso específico.

Teniendo en cuenta los datos proporcionados por estas mediciones, no hay ninguna diferencia sustancial en el tiempo de lectura y escritura entre el uso de WSL y Linux nativo.

## 8.4. Conclusión de los Benchmarks

La conclusión que se puede extraer de este capítulo es que, aunque WSL proporciona un alto grado de interoperabilidad con Windows, no tiene acceso completo a todas las características y funcionalidades del sistema operativo host. Por ejemplo, WSL no tiene acceso directo al hardware y puede tener algunas limitaciones en cuanto a la forma en que interactúa con la red y el sistema de archivos en comparación con un sistema Linux nativo.

Por ello, WSL no tiene acceso a toda la información del sistema Windows debido a que es una capa de compatibilidad diseñada para ejecutar binarios de Linux en Windows, y no un sistema operativo completo con acceso completo a todas las características y funcionalidades de la máquina donde está instalado. No obstante, con las pruebas realizadas se puede concluir que el rendimiento de WSL es casi el mismo que el de Linux nativo.