

# WSL: Guía de uso para el estudiante de la Escuela de Ingeniería de Fuenlabrada



*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: Guía de uso para el estudiante de la Escuela de Ingeniería de Fuenlabrada* es un manual que tiene como propósito brindar asistencia a los estudiantes de nuevo ingreso, permitiéndoles familiarizarse y sacar provecho de esta herramienta, enfocando su aplicación específicamente dentro del entorno universitario; 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.

Esta guía consta de seis capítulos, cada uno con sus respectivas secciones, que cubren la información más relevante que un estudiante sin experiencia previa en entornos Linux debe conocer.

**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:** El tercer capítulo se centra en la interoperabilidad de WSL con Windows y Linux. Explora las características compartidas y las diferencias, y las secciones del manual cubren temas como Usuarios en WSL, Compartir archivos y directorios y Ejecución de comandos y utilidades.

**Capítulo 4: Editores de texto:** Este capítulo tiene como objetivo enseñar a los estudiantes de nuevo ingreso en la Escuela de Ingeniería de Fuenlabrada las diferentes formas de editar archivos de texto. Se divide en dos subsecciones que tratan Editores en modo texto y IDEs (Entornos de Desarrollo Integrados).

**Capítulo 5: Conexión SSH:** El quinto capítulo se dedica a explicar qué es y cómo usar el protocolo SSH dentro de WSL. Además, incluye una sección sobre el uso de ALIAS para facilitar la conexión con las máquinas de laboratorio de la universidad. Las secciones cubiertas son ¿Qué es SSH?, Conexión a las máquinas Linux de la Universidad con WSL, Uso de SSH sin contraseña y Uso de ALIAS.

**Capítulo 6: Sincronización remota de archivos:** El sexto capítulo se enfoca en explicar cómo sincronizar archivos de forma remota para que los estudiantes puedan continuar su trabajo en los laboratorios de la Universidad desde otras ubicaciones. Se proponen alternativas como *Secure Copy (scp)*, *SSH Filesystem (sshfs)* y *FreeFileSync*. Este capítulo está diseñado para ayudar a usuarios con poca experiencia en entornos Linux a utilizar estas herramientas.

# Índice

1. Introducción.....	1
1.1. ¿Qué es WSL?.....	1
1.2. WSL frente a otras alternativas.....	1
2. Instalación de WSL.....	3
2.1. Prerrequisitos.....	3
2.2. Instalación.....	3
3. Interoperabilidad Windows-Linux.....	7
3.1. Usuarios.....	7
3.1.1. Funcionamiento.....	7
3.1.2. Restablecer contraseña.....	8
3.2. Compartir archivos y directorios.....	8
3.2.1. Acceder al sistema de ficheros de Linux (WSL) desde Windows.....	9
3.2.2. Acceder al sistema de ficheros de Windows desde Linux (WSL).....	9
3.3. Ejecución de comandos y utilidades.....	10
4. Editores de texto.....	11
4.1. Editores en modo texto.....	11
4.2. IDEs (Integrated Development Environment).....	11
5. Conexión ssh.....	13
5.1. ¿Qué es ssh?.....	13
5.2. Conexión a las máquinas Linux de la Universidad con WSL.....	13
5.2.1. Uso de ssh sin contraseña.....	15
5.2.2. Uso de ALIAS.....	18
6. Sincronización remota de archivos.....	20
6.1. Secure Copy (scp).....	20
6.1.1. ¿Qué es scp?.....	20
6.1.2. Copiar ficheros de la máquina de la Universidad a la máquina local.....	20
6.2. SSH Filesystem (sshfs).....	21
6.2.1. ¿Qué es sshfs?.....	21
6.2.2. Uso de sshfs.....	21
6.3. FreeFileSync.....	22
6.3.1. ¿Qué es FreeFileSync?.....	22
6.3.2. Uso de FreeFileSync.....	23

## Índice de figuras

Figura 2.2.1: Comando de instalación de WSL.....	4
Figura 2.2.2: Creación de usuario Linux.....	5
Figura 2.2.3: Instalación WSL finalizada.....	5
Figura 2.2.4: WSL lanzado desde PowerShell.....	6
Figura 3.2.1: Acceso a ficheros de WSL desde Windows.....	9
Figura 3.2.2: Acceso a ficheros de Windows desde WSL.....	10
Figura 3.3.1: Ejecución de comando Linux desde cmd de Windows.....	10
Figura 5.2.1: Verificación de autenticidad.....	14
Figura 5.2.2: Contraseña de usuario de los laboratorios.....	14
Figura 5.2.3: Conexión ssh establecida.....	15
Figura 5.2.4: Creación de claves pública-privada.....	16
Figura 5.2.5: Par de claves pública-privada.....	17
Figura 5.2.6: Contenido clave pública.....	17
Figura 5.2.7: Conexión ssh sin contraseña de usuario requerida.....	18
Figura 6.1.1: Copia vía scp completada.....	21
Figura 6.2.1: Montaje de directorio remoto vía sshfs.....	22
Figura 6.3.1: Pantalla principal de FreeFileSync.....	23
Figura 6.3.2: Selección de directorio local en FreeFileSync.....	24
Figura 6.3.3: Selección de directorio remoto en FreeFileSync.....	25
Figura 6.3.4: Iniciar proceso de sincronización bidireccional en FreeFileSync.....	25
Figura 6.3.5: Sincronización espejo en FreeFileSync.....	26

# 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. 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.

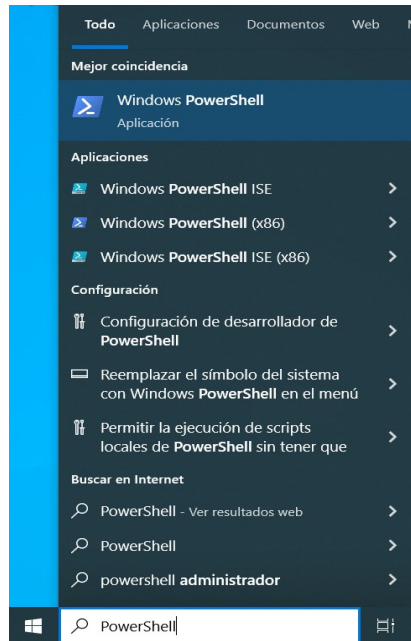
**1) Iniciar PowerShell o símbolo de sistema de Windows.** Para ello, dirigirse al explorador de Windows y escribir en la barra de búsqueda *PowerShell* o *cmd*. A continuación, hacer clic derecho sobre el programa en cuestión y dentro del menú de opciones que se despliega, clicar sobre *ejecutar como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



## Iniciar PowerShell



## Iniciar cmd (Símbolo del sistema de Windows)



**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](#).

## 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

#### 3.1.1. Funcionamiento

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 la distribución Windows, ha de hacerse directamente sobre Windows.

### 3.1.2. 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.

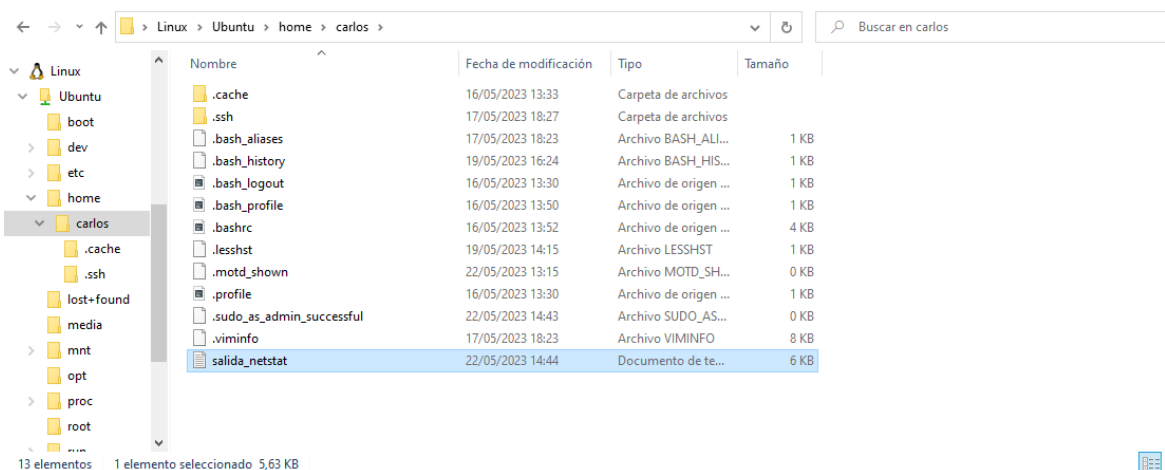


Figura 3.2.1: Acceso a ficheros de WSL desde Windows

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`. 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.

El acceso a ficheros que se encuentran en Windows desde WSL se consigue accediendo a los directorios que cuelgan del directorio `/mnt/c` siendo `c` el disco donde está instalado el sistema de ficheros de WSL.

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. Ejecución de comandos y utilidades

WSL permite ejecutar comandos y utilidades de Linux directamente desde el shell de Windows. Esto significa que se pueden utilizar comandos de Linux en la línea de comandos de Windows y viceversa.

Para ejecutar comandos de Linux en el intérprete de comandos de Windows (PowerShell y cmd), es necesario anteponer `wsl` al comando que se quiera ejecutar.

```
Símbolo del sistema
C:\Users\aleja\Desktop>ls
"ls" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
C:\Users\aleja\Desktop>wsl ls
carpeta_escritorio  desktop.ini
C:\Users\aleja\Desktop>_
```

Figura 3.3.1: Ejecución de comando Linux desde cmd de Windows

## 4. Editores de texto

Dentro de WSL, son varias las maneras que hay para editar un archivo de texto. Es posible realizar este trabajo íntegramente desde WSL sin la necesidad de tener que utilizar ningún programa instalado en Windows y, al mismo tiempo, también es posible realizarlo recurriendo a algún programa instalado en Windows que acceda a los ficheros de WSL. Por ello, existen dos alternativas diferentes en WSL para la edición de texto de ficheros.

### 4.1. Editores en modo texto

Los editores de texto en modo texto son herramientas que se ejecutan en la línea de comandos y ofrecen una interfaz basada en texto para editar archivos. Estos editores son muy potentes y altamente configurables.

Además, suelen ser los únicos disponibles en ordenadores a medio instalar, averiados, herramientas de rescate, etc.

La finalidad principal de los editores de texto en modo texto es proporcionar una forma eficiente de editar archivos de texto sin requerir una interfaz gráfica completa. Estos editores son muy populares entre los desarrolladores y administradores de sistemas debido a su velocidad, potencia y flexibilidad. Permiten realizar modificaciones rápidas en archivos de texto, navegar por ellos de forma eficiente y ejecutar comandos y scripts directamente desde el editor.

Usando editores en modo texto se puede hacer todo el trabajo de edición de texto dentro de WSL sin utilizar ningún programa adicional en Windows.

El editor *estándar de hecho* en Unix es *Vim*, o más bien, algún clon moderno, como *vim* o *neovim*. Para usarlo, basta con escribir dentro del terminal el comando `vi <nombre_archivo>`. Si el archivo ya existía, se abrirá para editarlo. Si no estaba creado, creará un fichero con ese nombre para su edición.

### 4.2. IDEs (Integrated Development Environment)

IDE es el acrónimo de *Integrated Development Environment*, que en español se traduce como Entorno de Desarrollo Integrado. Un IDE es una herramienta de software que proporciona un conjunto de características y funcionalidades para facilitar el desarrollo de software.

Un IDE típicamente incluye un editor de código fuente, herramientas de depuración, compiladores o intérpretes, y a menudo ofrece funcionalidades adicionales como resaltado de sintaxis, autocompletado, gestión de proyectos y control de versiones. Proporciona un entorno de trabajo completo y centralizado para desarrolladores, permitiéndoles escribir, probar y depurar código de manera más eficiente.



A diferencia de los editores en modo texto, los IDEs proporcionan un entorno más amigable y visualmente atractivo para desarrolladores y programadores. Ofrecen características avanzadas, como resaltado de sintaxis, autocompletado de código, depuración integrada, control de versiones y acceso a bibliotecas y frameworks específicos. Estos editores suelen ser más fáciles de usar para aquellos que están empezando en el desarrollo de software, ya que ofrecen una experiencia más visual e intuitiva.

Algunos de los IDEs más utilizados son *Visual Studio Code*, *Atom*, *Geany*, entre otros.

Por ello, es posible instalar cualquier IDE en Windows y configurarlo para que trabaje sobre cualquier fichero, incluyendo los ficheros de WSL. Es decir, es posible editar ficheros de WSL con programas instalados en Windows.

En el caso concreto de tener instalado el programa *Visual Studio Code* en la máquina Windows, es posible lanzarlo directamente desde el terminal de WSL mediante el comando

`code <nombre_archivo>`. Esto abrirá una sesión gráfica de dicho programa y, si el archivo ya existía, se abrirá para editarlo. Si no estaba creado, creará un fichero con ese nombre para su edición.

## 5. Conexión ssh

### 5.1. ¿Qué es ssh?

ssh (Secure Shell) es un protocolo de red que permite a los usuarios acceder y administrar de forma segura en modo texto mediante comandos un sistema remoto a través de una red.

Este protocolo de red brinda la posibilidad de conectarse a una *máquina remota* (máquina con la que se quiere establecer la conexión) y gestionarla desde la *máquina local* (máquina en la que se está trabajando).

Sin embargo, no a todas las máquinas se puede llegar vía ssh (por lo menos en un inicio). Para poder establecer conexión con una máquina remota, es necesario cumplir con una serie de requisitos:

- **Tener acceso a un sistema operativo compatible:** ssh es compatible con sistemas operativos Unix (Linux, macOS) o Windows.
- **Tener una conexión de red:** ssh utiliza una conexión de red para establecer la comunicación entre el cliente y el servidor. Por ello, hay que tener una conexión de red activa y funcional.
- **Tener un cliente y servidor ssh configurado:** El servidor ssh debe estar instalado y configurado en el sistema remoto al que se quiere acceder y el cliente ssh debe estar instalado y configurado en la máquina desde la que se quiere establecer conexión. Por defecto, los sistemas Unix suelen tener instalado tanto el cliente como el servidor ssh mientras que en los Windows no es habitual tenerlo. Por ello, sería necesario instalar software adicional en la máquina.
- **Tener las credenciales de acceso:** Será necesario tener las credenciales de acceso al servidor remoto (nombre de usuario y contraseña).
- **Tener permisos y autorizaciones adecuados:** Hay que tener los permisos y autorizaciones pertinentes para acceder al servidor remoto a través de ssh. Esto puede requerir que la cuenta de usuario a la que se desea acceder esté configurada correctamente en el servidor y que tengas los privilegios adecuados.

### 5.2. Conexión a las máquinas Linux de la Universidad con WSL

WSL, a diferencia de Windows donde por norma general hay que instalar software adicional para poder utilizar ssh, cuenta por defecto con los paquetes necesarios para establecer conexiones ssh con máquinas remotas. Sin embargo, si se quisiera acceder a una máquina Windows que tiene WSL instalado vía ssh, sería necesario hacer una serie de configuraciones previas ya que no cuenta con esta característica por defecto.

Por ello, una máquina Windows puede utilizar ssh para conectarse a una máquina externa gracias a WSL sin tener que hacer nada más que instalar esta herramienta en la

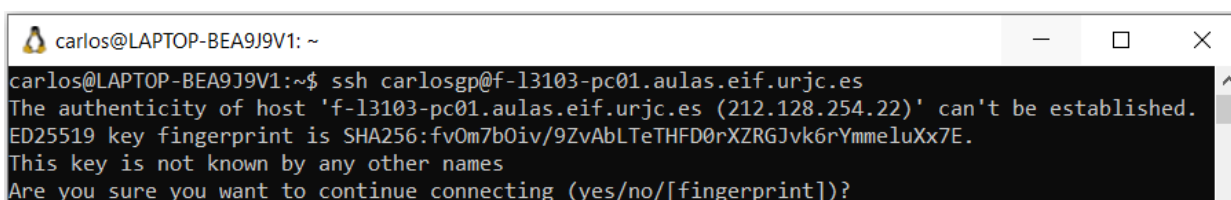
máquina.

En el caso particular de la EIF, algunas de las máquinas de los laboratorios están configuradas para poder acceder a ellas vía ssh desde una red externa.

Para saber cuáles son las máquinas a las que se puede acceder, es necesario consultar el parte de guerra de los laboratorios (<https://labs.eif.urjc.es/parte/>).

El acceso por ssh a las máquinas de la Universidad con WSL se hace de manera idéntica a como se haría con un Linux nativo.

El primer paso para establecer conexión vía ssh con las máquinas de la Universidad es ejecutar el comando `ssh usuario@maquina.aulas.eif.urjc.es` en WSL donde `usuario` es el nombre de usuario de la cuenta de los laboratorios y `maquina` una de las máquinas del parte de guerra. En este ejemplo, la máquina utilizada es `f-13103-pc01`. Al nombre de la máquina, hay que añadirle la extensión (dominio) de la universidad: `.aulas.eif.urjc.es`



```
carlos@LAPTOP-BEA9J9V1: ~
carlos@LAPTOP-BEA9J9V1:~$ ssh carlosgp@f-13103-pc01.aulas.eif.urjc.es
The authenticity of host 'f-13103-pc01.aulas.eif.urjc.es (212.128.254.22)' can't be established.
ED25519 key fingerprint is SHA256:fv0m7b0iv/9ZvAbLTeTHFD0rXZRGJvk6rYmmeluXx7E.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

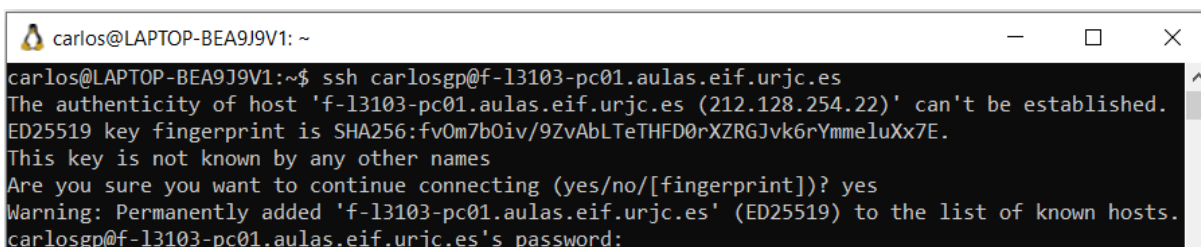
*Figura 5.2.1: Verificación de autenticidad*

Dado que es la primera vez que se está intentando acceder vía ssh desde la máquina local a una de las máquinas del laboratorio, el terminal mostrará el anterior mensaje por pantalla.

El sistema verifica que el servidor sea realmente quien dice ser dado que no ha habido ninguna conexión ssh anterior a esta. La identidad se verifica mediante una clave única generada para el servidor.

La clave se muestra como una huella digital (fingerprint), que es una secuencia de caracteres única que identifica al servidor. Si se necesita la certeza absoluta de que esta máquina es quien dice ser, se debería comprobar esta huella digital por un medio seguro, alternativo.

En este caso, se escribe `yes` a la anterior pregunta ya que es cierto que realmente se trata de una de las máquinas del laboratorio. En sucesivas ocasiones, este mensaje no se volverá a mostrar.



```
carlos@LAPTOP-BEA9J9V1: ~
carlos@LAPTOP-BEA9J9V1:~$ ssh carlosgp@f-13103-pc01.aulas.eif.urjc.es
The authenticity of host 'f-13103-pc01.aulas.eif.urjc.es (212.128.254.22)' can't be established.
ED25519 key fingerprint is SHA256:fv0m7b0iv/9ZvAbLTeTHFD0rXZRGJvk6rYmmeluXx7E.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'f-13103-pc01.aulas.eif.urjc.es' (ED25519) to the list of known hosts.
carlosgp@f-13103-pc01.aulas.eif.urjc.es's password:
```

*Figura 5.2.2: Contraseña de usuario de los laboratorios*

A continuación, se pide que se escriba la contraseña del usuario de la máquina a la que se quiere acceder (contraseña del usuario de los laboratorios). Tras escribirla correctamente, la conexión habrá sido establecida con éxito.



```
carlos@LAPTOP-BEA9J9V1: ~  
carlos@LAPTOP-BEA9J9V1:~$ ssh carlosgp@f-l3103-pc01.aulas.eif.urjc.es  
The authenticity of host 'f-l3103-pc01.aulas.eif.urjc.es (212.128.254.22)' can't  
be established.  
ED25519 key fingerprint is SHA256:fv0m7b0iv/9ZvAbLTeTHFD0rXZRGJvk6rYmmeluXx7E.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'f-l3103-pc01.aulas.eif.urjc.es' (ED25519) to the list  
of known hosts. carlosgp@f-l3103-pc01.aulas.eif.urjc.es's password:  
  
┌───┴───┐  
│ Bienvenido al host f-l3103-pc01.aulas.eif.urjc.es  
│ de los Laboratorios Docentes de la EIF (URJC) │  
└───┬───┘  
┌───┴───┐  
│ El uso de esta cuenta conlleva la aceptacion de las normas de  
│ uso de estos Laboratorios, expuestas en la pagina web:  
│ https://labs.eif.urjc.es/normas  
│  
│ No hacerlo puede suponer el cierre PERMANENTE de la cuenta.  
│  
│ Recuerda que esta maquina se reinicia automaticamente todos los dias  
│ a las 06:00am.  
│  
│ Si no estas autorizado a usar estas maquinas  
│ DEBES DESCONECTAR INMEDIATAMENTE.  
│ Todos los accesos son monitorizados y AUDITADOS.  
│  
│ Pagina Web del Laboratorio: https://labs.eif.urjc.es  
│ Si necesitas ayuda, mandanos un mail  
│ a esta direccion: sopORTE-lab@gsyc.urjc.es  
│  
│ Have Fun!  
└───┬───┘  
Last login: Thu Jun 15 17:17:08 2023 from 92.172.119.151  
carlosgp@f-l3103-pc01:~$
```

Figura 5.2.3: Conexión ssh establecida

Por cada vez que se realice una conexión ssh a esta máquina, será necesario volver a introducir la contraseña del usuario de los laboratorios; sin embargo, este comportamiento puede cambiarse para que no sea necesario escribirla.

### 5.2.1. Uso de ssh sin contraseña

Si no se quiere tener la necesidad de escribir la contraseña cada vez que se realice una conexión ssh a las máquinas de la Universidad (o a cualquier otra máquina), es necesario tener otro método de autenticación para verificar que quien está intentando establecer la conexión es quien dice ser.

En el contexto de ssh, se puede utilizar un par de claves pública-privada para establecer una autenticación sin contraseña. En lugar de introducir una contraseña cada vez que se intenta acceder a un servidor remoto, se puede generar un par de claves pública-privada

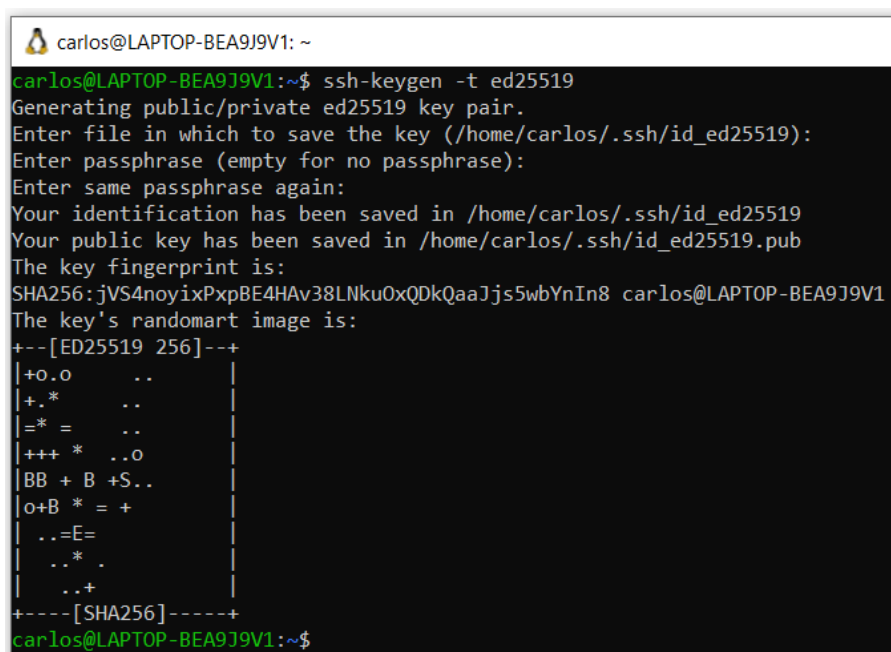
en el cliente ssh (en WSL) y agregar la clave pública correspondiente al archivo de autorización del servidor remoto.

Cuando se intenta establecer una conexión ssh, el cliente utiliza su clave privada para demostrar su identidad al servidor remoto. Si la clave privada coincide con la clave pública almacenada en el servidor, se permite el acceso sin contraseña. Esto proporciona un nivel adicional de seguridad, ya que no es necesario transmitir ni almacenar contraseñas en el proceso de autenticación.

Existen varios algoritmos criptográficos que se utilizan para generar pares de claves pública-privada. En este caso, el par de claves va a ser creado utilizando el algoritmo Ed25519.

- **Crear par de claves pública-privada**

Desde el terminal de WSL, se crea el par de claves con el comando `ssh-keygen -t ed25519`



```
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 5.2.4: Creación de claves pública-privada

Tras haber introducido el comando, se pedirá que se introduzca la ruta y el nombre del fichero donde guardar el par de claves y una contraseña para proteger la clave privada. En este caso, no se ha introducido nada por lo que en cada pregunta se ha presionado la tecla *enter* (salto de línea) para utilizar los parámetros por defecto. Por ende, las contraseñas se han guardado dentro del directorio `/home/<usuario>/.ssh/` con el nombre por defecto de `id_ed25519.pub` para la clave pública y `id_ed25519` para la clave privada.

```
carlos@LAPTOP-BEA9J9V1: ~/.ssh
carlos@LAPTOP-BEA9J9V1:~$ cd .ssh/
carlos@LAPTOP-BEA9J9V1:~/.ssh$ ls
id_ed25519 id_ed25519.pub known_hosts known_hosts.old
carlos@LAPTOP-BEA9J9V1:~/.ssh$
```

Figura 5.2.5: Par de claves pública-privada

- **Copiar el contenido de la clave pública en la máquina de la Universidad (remota).**

A continuación, hay que visualizar el contenido de la **clave pública** y copiarlo dentro del fichero `.ssh/authorized_keys` de la **máquina del laboratorio**.

Para visualizar el contenido del fichero que contiene la clave pública, se puede ejecutar la orden `cat id_ed25519.pub` dentro del directorio donde esté dicho fichero.

```
carlos@LAPTOP-BEA9J9V1: ~/.ssh
carlos@LAPTOP-BEA9J9V1:~/.ssh$ ls
id_ed25519 id_ed25519.pub known_hosts known_hosts.old
carlos@LAPTOP-BEA9J9V1:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIM1IjUsjzyS2T86BFBNI/KduI1oSLZA3Xltkm+8fI/kF carlos@LAPTOP-BEA9J9V1
carlos@LAPTOP-BEA9J9V1:~/.ssh$
```

Figura 5.2.6: Contenido clave pública

- **Pegar el contenido de la clave pública en la máquina del laboratorio.**

En primer lugar, hay que estar conectado a la máquina de la Universidad. Por ejemplo, vía ssh (ver apartado 5.2).

Una vez se esté conectado, editar el fichero `.ssh/authorized_keys` con el IDE que se prefiera para pegar ahí el contenido de la clave pública.

Tras haberlo copiado correctamente, las próximas veces que se quiera establecer conexión vía ssh a esa máquina, no será necesario escribir la contraseña de usuario.



1. Reiniciar el terminal donde se está trabajando.
2. Ejecutar la orden `source ~/.bashrc`

Para agilizar el establecimiento de conexión a la máquina de la Universidad, es útil tener un alias similar a este en el fichero `.bash_aliases`:

```
alias laboratorio='ssh carlosgp@f-13103-pc01.aulas.eif.urjc.es'
```

siendo `carlosgp` el nombre del usuario. De esta manera, al escribir el comando `laboratorio` se ejecutará directamente todo el comando de `ssh`.



## 6. Sincronización remota de archivos

Gracias a WSL, se pueden utilizar diferentes herramientas propias de Linux dentro de Windows que, sin WSL, en la mayoría de los casos no sería posible.

En este capítulo, se explorarán diversas herramientas y técnicas que permitirán establecer conexiones seguras y sincronizar archivos entre una máquina remota y una máquina local. En concreto, se particularizarán sus usos para el caso de acceso remoto y sincronización de archivos con las máquinas de los laboratorios de la EIF.

### 6.1. Secure Copy (scp)

#### 6.1.1. ¿Qué es scp?

scp (Secure Copy Protocol) es un protocolo de transferencia de archivos seguro basado en ssh. Se utiliza para transferir archivos entre un cliente y un servidor de manera segura y cifrada.

El funcionamiento de scp se basa en una conexión ssh establecida entre el cliente y el servidor por lo que es necesario tenerlo previamente configurado. Utiliza la autenticación y encriptación proporcionada por ssh para garantizar la seguridad de la transferencia de archivos.

#### 6.1.2. Copiar ficheros de la máquina de la Universidad a la máquina local

Con scp es posible copiar archivos de una máquina remota a una máquina local.

Si se quisiera tener copiada en una máquina local alguna información que esté en los laboratorios de la Universidad, basta con utilizar el siguiente comando:

```
scp -r usuario@maquina.aulas.eif.urjc.es:<dir_remoto>  
<dir_local>
```

donde `-r` es una opción del comando que significa *iterative* que sirve para copiar un directorio y todos sus subdirectorios, `dir_remoto` es la ruta donde se encuentra en la máquina remota lo que se quiere copiar y `dir_local` la ruta donde se copiará la información en la máquina local.

Si por ejemplo se quisiera copiar el contenido del directorio Documentos de la máquina de la Universidad al directorio actual de la máquina local habría que ejecutar el siguiente comando:

```
scp -r  
usuario@f-13103-pc01.aulas.eif.urjc.es:/home/alumnos/usuario/Documentos .  
donde el punto final significa directorio actual de la máquina local.
```

```
carlos@LAPTOP-BEA9J9V1: ~  
carlos@LAPTOP-BEA9J9V1:~$ pwd  
/home/carlos  
carlos@LAPTOP-BEA9J9V1:~$ ls  
carlos@LAPTOP-BEA9J9V1:~$ scp -r carlosgp@f-13103-pc01.aulas.eif.urjc.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 6.1.1: Copia vía scp completada

Como scp está basado en ssh, en este caso no ha sido necesario teclear la contraseña del usuario de la Universidad porque en el capítulo anterior se configuró para que no lo hiciera. Si no se hubiera configurado el uso de ssh sin contraseña (apartado 5.2.1), ahora habría sido necesario escribirla.

## 6.2. SSH Filesystem (sshfs)

### 6.2.1. ¿Qué es sshfs?

sshfs (Secure Shell File System) es una herramienta que permite montar sistemas de archivos remotos sobre ssh. Proporciona una forma segura y conveniente de acceder y gestionar archivos ubicados en un servidor remoto como si estuvieran presentes en el sistema de archivos local.

sshfs, al igual que scp, es una alternativa más para trabajar con los datos que están en una máquina remota. Sin embargo, al usar scp, cada vez que se quiera acceder a un fichero remoto es necesario copiarlo mientras que mediante sshfs, una vez que se haya montado el sistema de ficheros remoto, se puede trabajar sin pasos adicionales, exactamente igual que si el fichero estuviese en la máquina local.

Para poder usarlo, es necesario instalar el paquete. Para ello, primero hay que tener los repositorios de la distribución actualizados (`sudo apt-get update` && `sudo apt-get upgrade`) y ejecutar el comando de instalación del paquete (`sudo apt-get install sshfs`).

### 6.2.2. Uso de sshfs

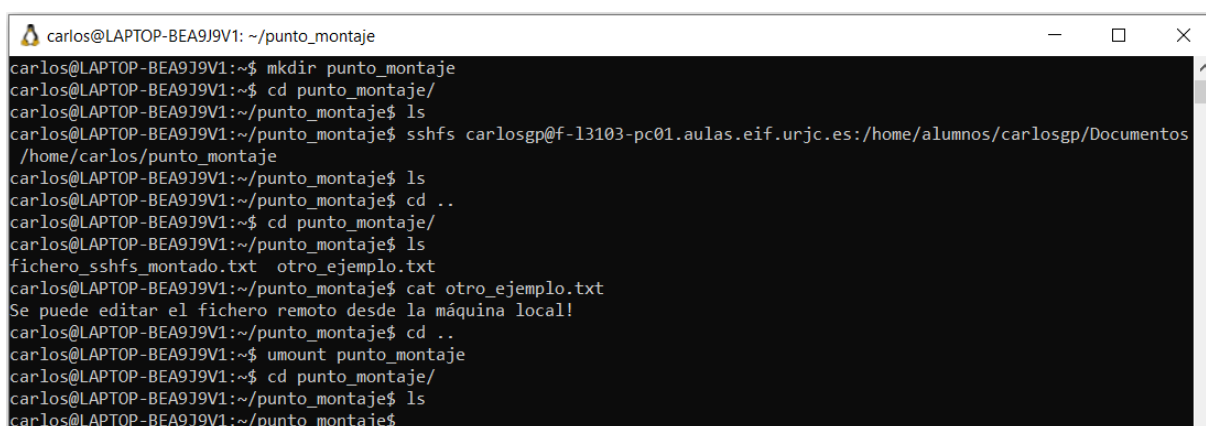
El comando para poder montar información de una máquina remota a la máquina local usando sshfs es `sshfs usuario@maquina.aulas.eif.urjc.es:<dir_remoto> <punto_montaje>` donde `dir_remoto` es la ruta donde se encuentra en la máquina remota lo que se quiere copiar y `punto_montaje` es la ruta donde se montará la información en la máquina local, por ejemplo, un directorio *labs* de la carpeta personal del usuario. Se debe observar que el segundo parámetro, el punto de montaje, debe ser un directorio preexistente, normalmente vacío, que será reemplazado por el directorio remoto.

Si se quisiera montar en la máquina local el directorio `~/Documentos` de la máquina remota en un directorio de la máquina local, el comando a ejecutar sería el siguiente:

```
sshfs
usuario@maquina.aulas.eif.urjc.es:/home/alumnos/usuario/Documentos <punto_montaje>
```

Una vez que se haya terminado de trabajar, el directorio remoto montado se puede desmontar con el comando `umount <directorio_local>` siendo `directorio_local` la ruta del directorio local donde se montó el sistema de archivos remoto.

Un ejemplo de ejecución completo sería el siguiente:



```
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.urjc.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 6.2.1: Montaje de directorio remoto vía sshfs

## 6.3. FreeFileSync

### 6.3.1. ¿Qué es FreeFileSync?

FreeFileSync es una aplicación de software libre de sincronización y respaldo de archivos que permitirá mantener archivos sincronizados entre diferentes máquinas. FreeFileSync puede utilizarse para establecer tareas de sincronización, comparar y copiar archivos de manera eficiente y mantener los datos actualizados y respaldados en las máquinas con las que se esté trabajando. Además, FreeFileSync permite comparar dos directorios siendo capaz de detectar las diferencias entre ambos y propagarlas de forma que ambos directorios acaben teniendo el mismo contenido. Los directorios pueden estar en la máquina local o en una máquina remota accesible a través de FTP o SFTP. También soporta Google Drive. Es una aplicación libre y gratuita disponible para Microsoft Windows, Linux y macOS.

Al ser una aplicación externa, para poder utilizarla primero será necesario descargarla (<https://freefilesync.org/>).

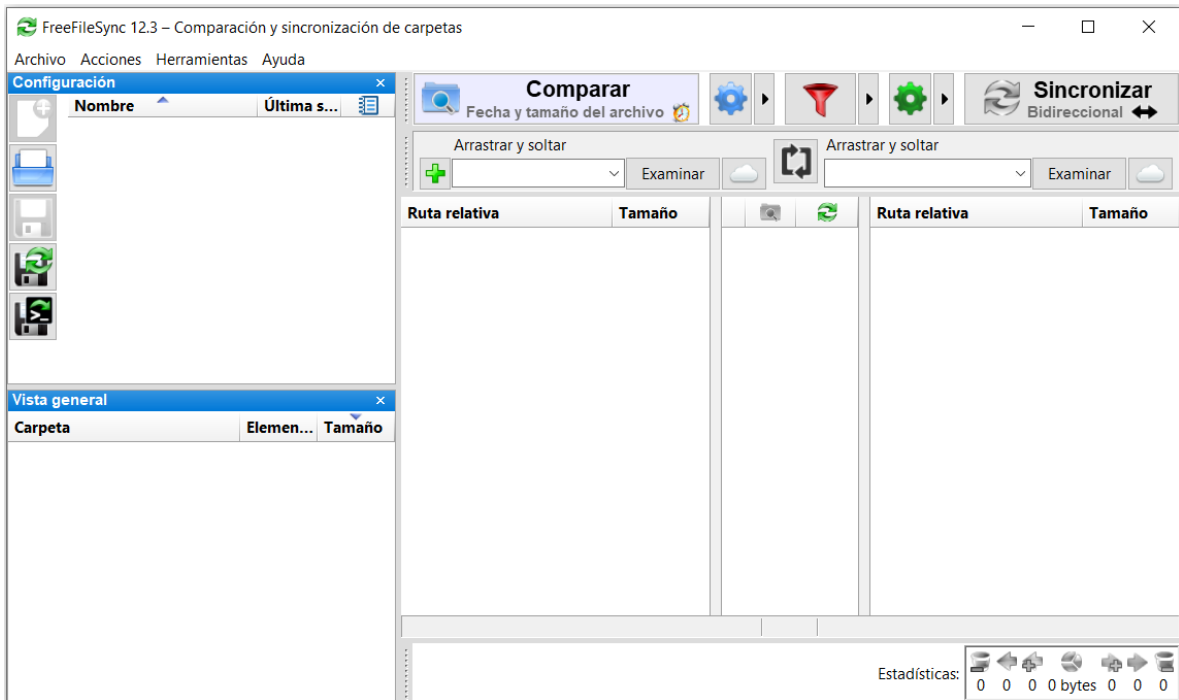


Figura 6.3.1: Pantalla principal de FreeFileSync

### 6.3.2. Uso de FreeFileSync

Con FreeFileSync se realizarán principalmente dos tipos de sincronización:

- Bidireccional
- Espejo

En ambos casos, antes de sincronizar se tendrán dos directorios que normalmente serán parecidos pero con algunas diferencias. Después de sincronizar, ambos directorios serán idénticos.

Ambos directorios podrán estar en cualquier lugar, es decir, pueden estar en el mismo disco de la misma máquina, en otro disco de otra máquina o incluso en el mismo disco pero otra máquina (como en el caso de los laboratorios).

#### 1. Sincronización en modo bidireccional.

En la sincronización bidireccional, todas las novedades del primer directorio se propagarán al segundo (ficheros nuevos, modificados o borrados, directorios nuevos, modificados o borrados) y también todas las novedades del segundo directorio se propagarán al primero. Tras la sincronización, ambos directorios serán idénticos.

Ejemplo de uso: el primer directorio es el directorio de trabajo en WSL (máquina local) y el segundo directorio es el directorio de trabajo en la máquina remota (máquina de los laboratorios).

En primer lugar, es necesario seleccionar la ruta donde se encuentra el directorio local (flecha roja izquierda) y el directorio remoto (flecha roja derecha). Para seleccionar el directorio de la máquina remota (directorío de los laboratorios), primero hay que pulsar

sobre el icono de la nube e iniciar sesión con la cuenta del laboratorio utilizando el protocolo SFTP.

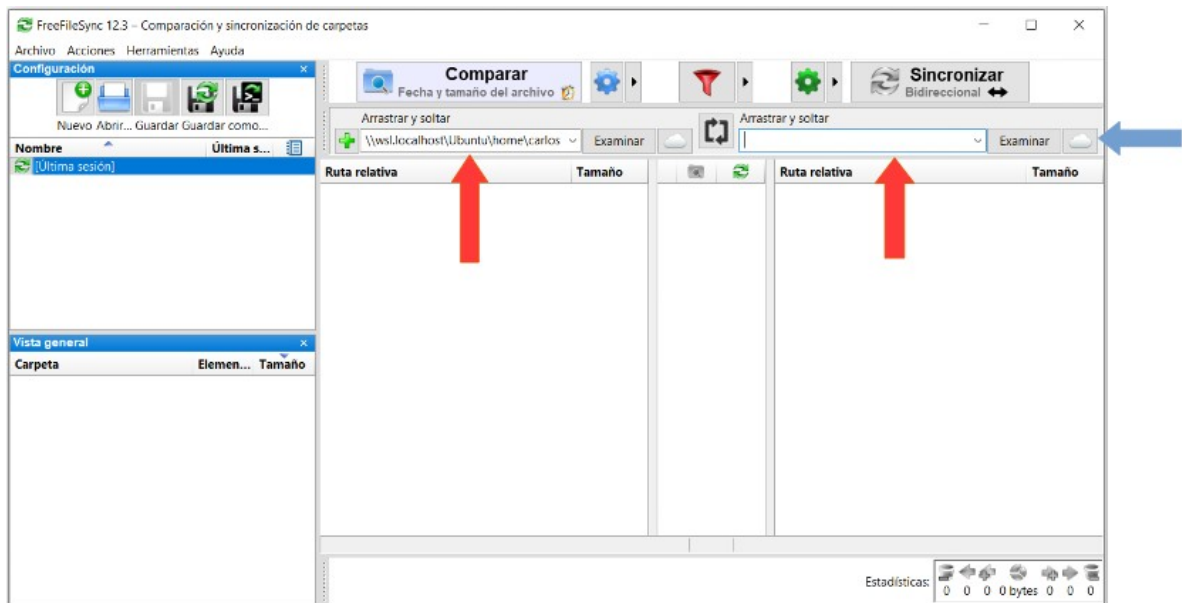


Figura 6.3.2: Selección de directorio local en FreeFileSync

Tras seleccionar el icono de la nube, habrá que introducir los datos que referencian a la máquina de los laboratorios.

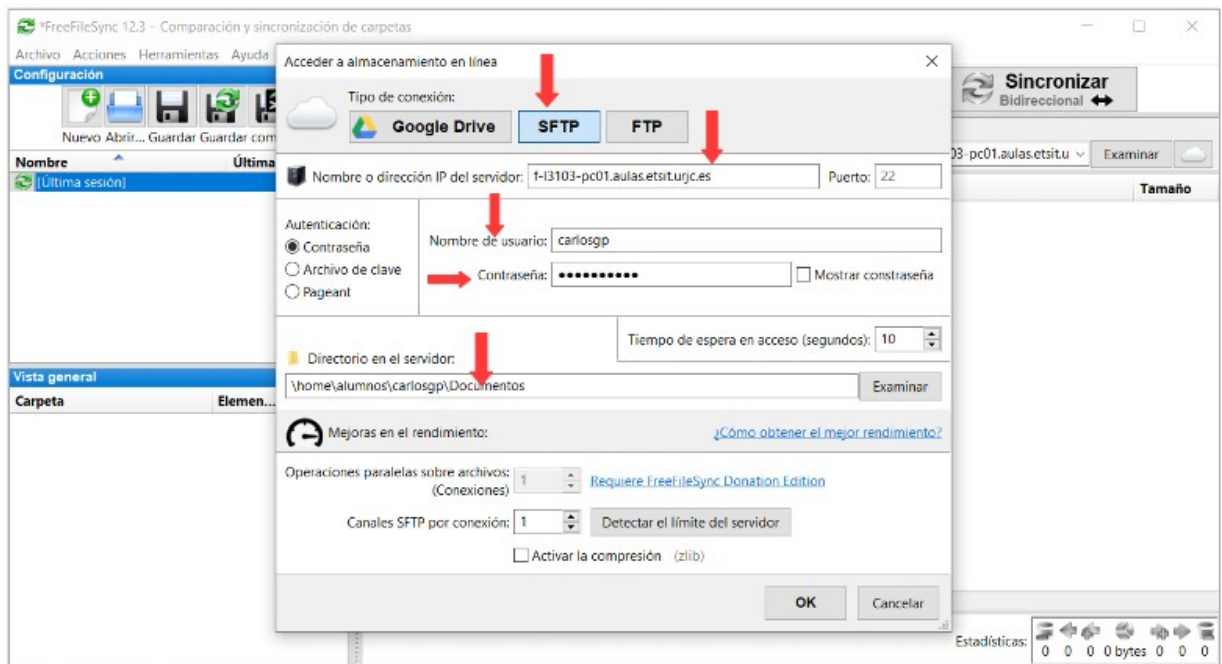


Figura 6.3.3: Selección de directorio remoto en FreeFileSync

Tras ello, clicar el botón OK para confirmar los datos y, a continuación, clicar sobre el botón de sincronizar (botón de la fila superior a la derecha del todo). Tras pulsarlo, se pondrá de color verde y aparecerá una ventana emergente preguntando por la inicialización de la sincronización.

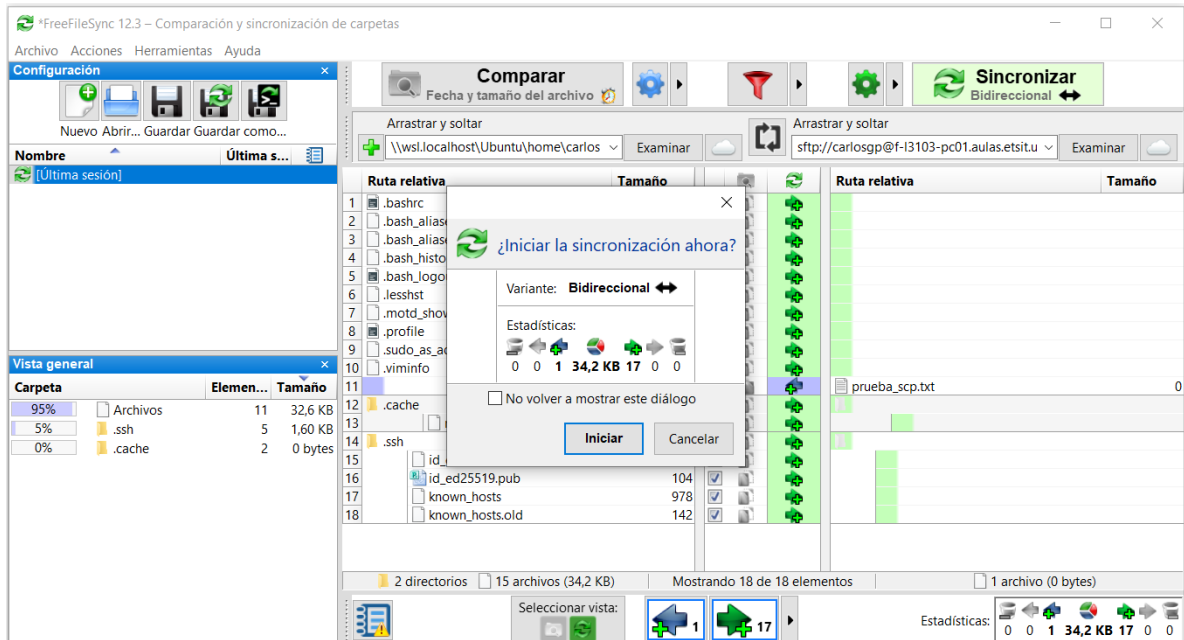


Figura 6.3.4: Iniciar proceso de sincronización bidireccional en FreeFileSync

Tras haber pulsado sobre el botón de *iniciar*, se iniciará el proceso de sincronización. Cuando haya finalizado, se podrá comprobar cómo en el directorio de la máquina local hay la misma información que en el directorio de la máquina remota (máquina de los laboratorios).

## 2. Sincronización en modo espejo.

En la sincronización espejo, al igual que en la sincronización en modo bidireccional, se tienen dos directorios: el principal y la copia; después de la sincronización, ambos directorios serán idénticos.

Por ello, todas las novedades del principal se llevarán a la copia; en el directorio copia, no debería haber novedades ya que, si las hay, serán ignoradas y destruidas. Un caso de uso típico de esta variante de sincronización es cuando se tiene un directorio de trabajo y se quiere hacer una copia de seguridad de él.

Para llevar a cabo esta sincronización, hay que seguir los mismos pasos que en la sincronización bidireccional cambiando el tipo de sincronización. Para ello, hay que clicar sobre la rueda dentada de color verde. A continuación, sobre la pantalla emergente que aparece, seleccionar la variante *espejo*. Tras ello, clicar sobre el botón *OK* e iniciar la sincronización de manera idéntica al proceso anterior.

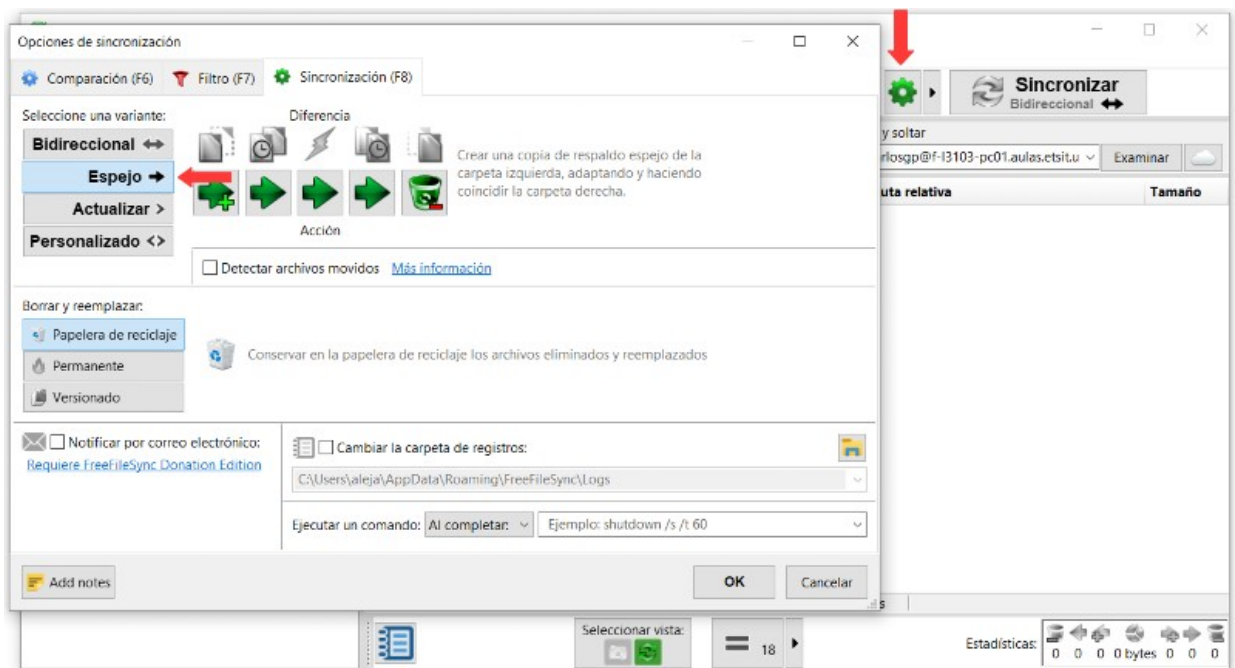


Figura 6.3.5: Sincronización espejo en FreeFileSync