

Computación Paralela (CPA)

Prácticas

Presentación

J. Javier Ibáñez González

Despacho D109 –

DSIC (1F)



Tutorías: bajo demanda

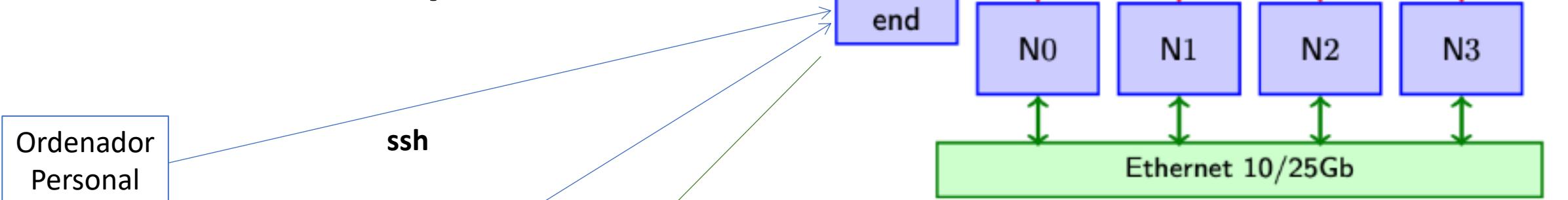
e-mail: jjibanez@upv.es

<http://personales.upv.es/jjibanez>

Prácticas de CPA

- P1. Paralelización con OpenMP
 - Sesión 1: Ejercicios básicos
 - Sesión 2: Procesamiento de imágenes
 - Sesión 3: Números primos
 - Sesión 4: Simulación astronómica de cometas
 - Sesión 5: Evaluación de P1 (Prueba tipo test S1-S4: 15% de la nota final)
- P2. Paralelización con MPI
 - Sesión 6: Ejercicios básicos
 - Sesión 7: Comunicación punto a punto con MPI
 - Sesión 8: Comunicación colectiva
 - Sesión 9: Tipos de datos
 - Sesión 10: Evaluación de P2 (Prueba tipo test S6-S9 : 15% de la nota final)

Cluster de prácticas Kahan



El cluster consta de un ordenador **front-end** y de 4 nodos. Cada nodo consta de

- 1 procesador AMD EPYC 7551P de 32 núcleos físicos (64 virtuales)
- 64GB de memoria
- Disco SSD de 240GB
- Ethernet 10/25Gb 2-port 622FLR -SFP28

- En el **front-end** de **Kahan** (acceso mediante **ssh**):
 - Los códigos ejecutables deben estar almacenados en un directorio del **front-end**
 - Se pueden hacer cosas muy básicas: edición, compilación y ejecuciones poco costosas
 - Las ejecuciones sobre el cluster se realizan desde el **front-end** mediante el sistema de colas **SLURM**, usando pruebas con un elevado coste computacional
- Documentación:

<http://personales.upv.es/jroman/kahan.html>

Ordenador Personal

ssh

ssh

<https://polilabs.upv.es>

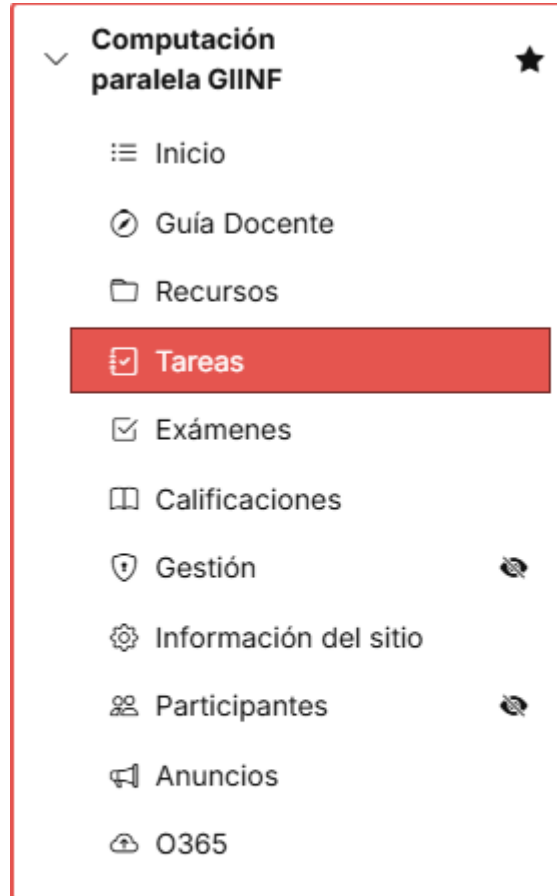
Ordenador del laboratorio

Disco W (visible desde **Polilabs** y **Front-end**)

- En este disco se copian los ficheros de las prácticas procedentes de **Tareas de Poliformat**
- Se pueden editar los códigos, compilar y realizar ejecuciones poco costosas para ver su correcto funcionamiento
- Los nombres de los directorios en **disco W** no pueden contener espacios en blanco

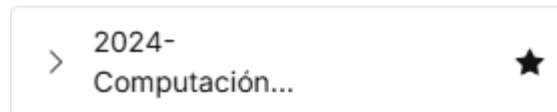
Material de prácticas

- El material de las prácticas (boletines y códigos fuente) se encuentra en Poliformat/Tareas



Computación paralela GIINF

- Inicio
- Guía Docente
- Recursos
- Tareas**
- Exámenes
- Calificaciones
- Gestión
- Información del sitio
- Participantes
- Anuncios
- 0365



> 2024- Computación...

turnos si fuese necesario (se avisará). Se podrá

.....


Practice 1, based on the **OMP**, will be assessed materials developed by the student will be avail:


.....

Pràctica 1, basada en **OMP** que serà avaluada material de pràctiques desenvolupat per l'alumr

Recursos adicionales para la tarea

 [cpa-p1-e.pdf](#) (2 MB; 23 sept 2025 13:54)

 [cpa-p1.pdf](#) (2 MB; 23 sept 2025 13:54)

 [material-p1.tgz](#) (4 MB; 18 sept 2025 11:38)

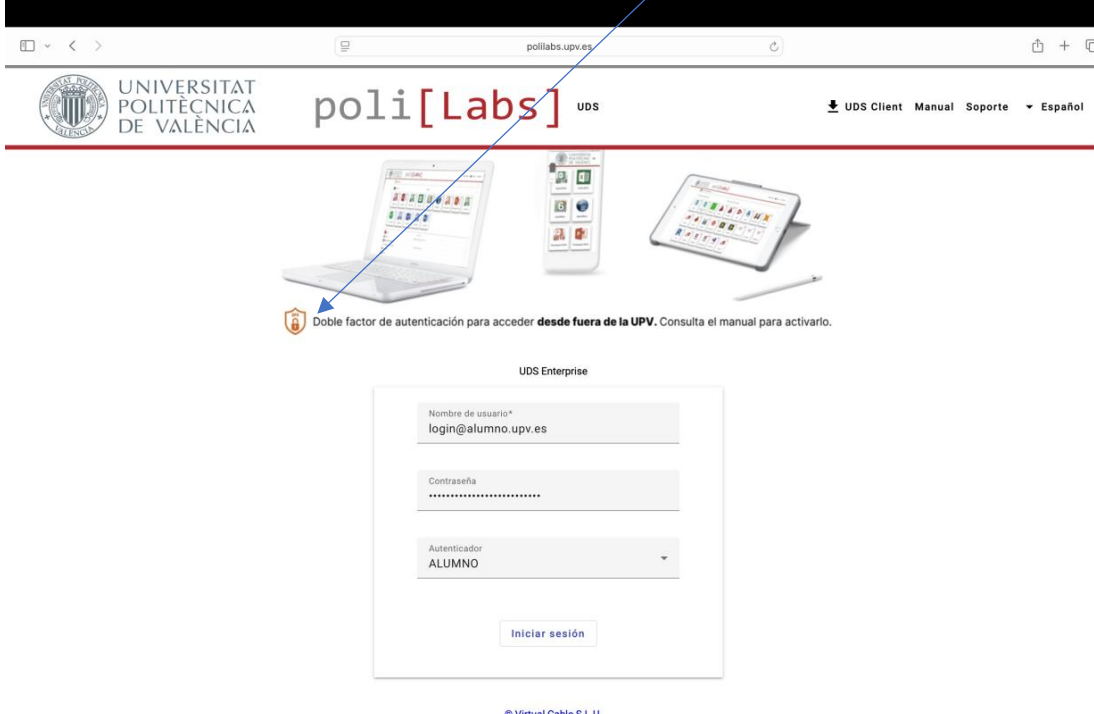
 [cpa-p1-v.pdf](#) (2 MB; 23 sept 2025 13:54)

Trabajo en el laboratorio

- Hay dos posibilidades:
 - Usar el ordenador del laboratorio o con un portátil, usando **Polilabs**:
<https://polilabs.upv.es>
 - Usar un portátil con gcc y OpenMPI instalados
 - Después, se puede trabajar:
 - primero con la máquina virtual/ ordenador laboratorio/portátil y después con Kahan
 - directamente con Kahan

1. Acceso remoto mediante Polilabs

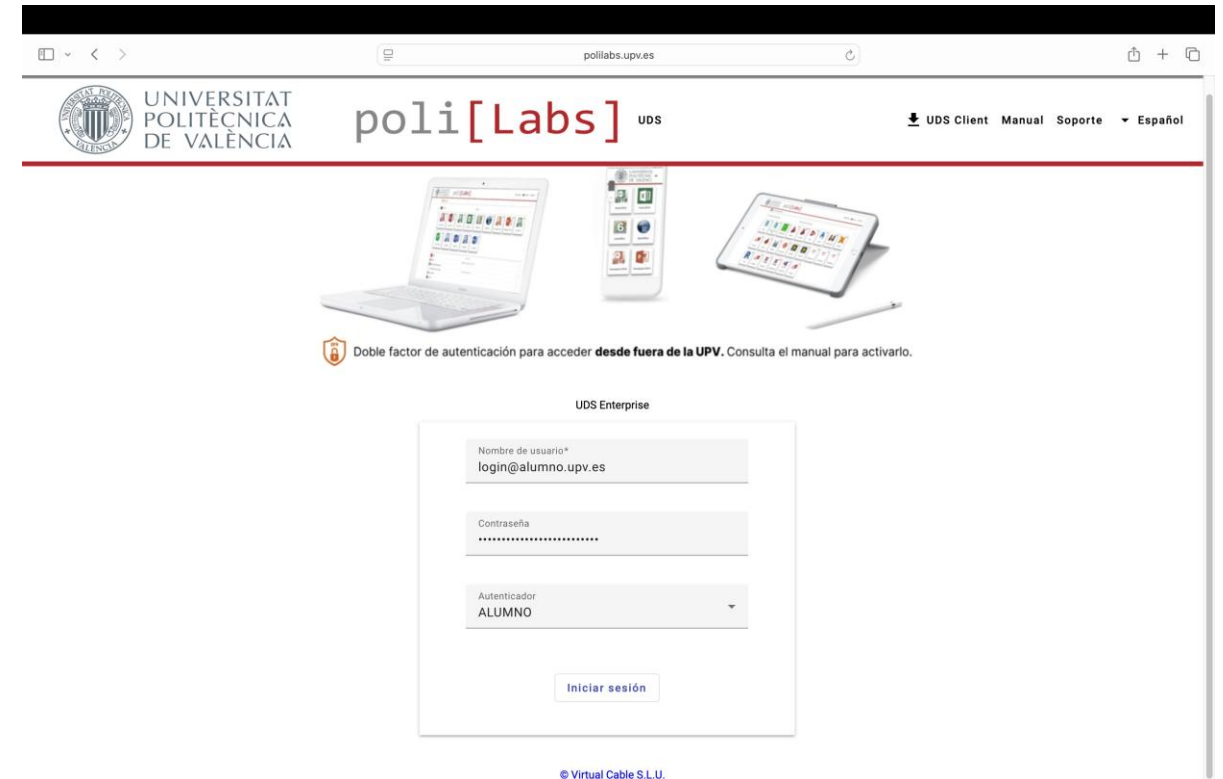
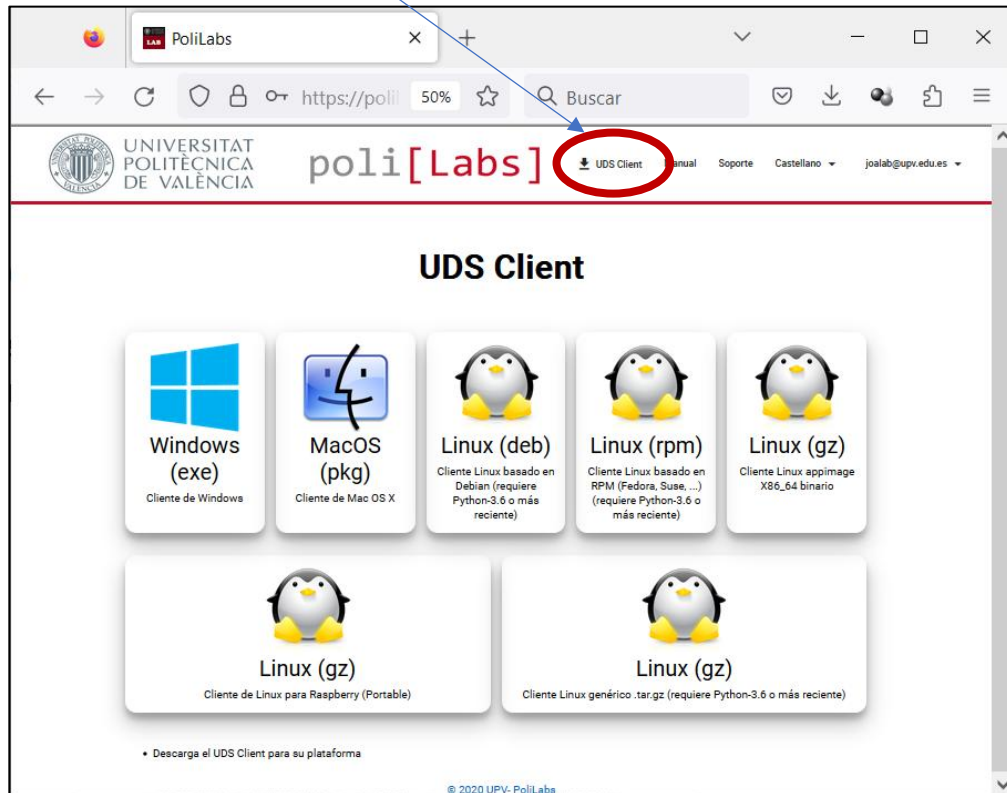
- El servicio está accesible remotamente desde cualquier equipo dentro y fuera de la UPV desde la dirección <https://polilabs.upv.es>.
- Si estás fuera de la UPV, se accede mediante un doble **factor de autenticación**. Tendrás que tener instalado la APP **Authenticator** de Microsoft. Cuando te vayas a autenticar tendrás que introducir un código, el cual aparece en la APP **Authenticator**



The screenshot shows the Polilabs website interface. At the top, there is a navigation bar with the UPV logo, the text "UNIVERSITAT POLITÈCNICA DE VALÈNCIA", the "poli[Labs] UDS" logo, and links for "UDS Client", "Manual", "Soporte", and "Español". Below the navigation bar, there are images of a laptop, a smartphone, and a tablet, all displaying the Polilabs interface. A blue arrow points from the text "factor de autenticación" in the second bullet point to the laptop image. Below the images, there is a warning icon and text: "Doble factor de autenticación para acceder desde fuera de la UPV. Consulta el manual para activarlo." Below this warning, there is a login form titled "UDS Enterprise" with the following fields: "Nombre de usuario*" with the value "login@alumno.upv.es", "Contraseña" with masked characters, and "Autenticador" with the value "ALUMNO". There is an "Iniciar sesión" button at the bottom of the form. At the very bottom of the page, there is a small copyright notice: "© Virtual Cable S.L.U."

1. Acceso remoto desde Windows o Linux

- La primera vez que accedamos al servicio, será necesaria la instalación del **conector** llamado **UDS Client**.
- **Descargamos** el cliente Windows o Linux y lo **instalamos**.
- Volvemos a la página anterior e introducimos nuestro **login** y **password** de la **UPV**.



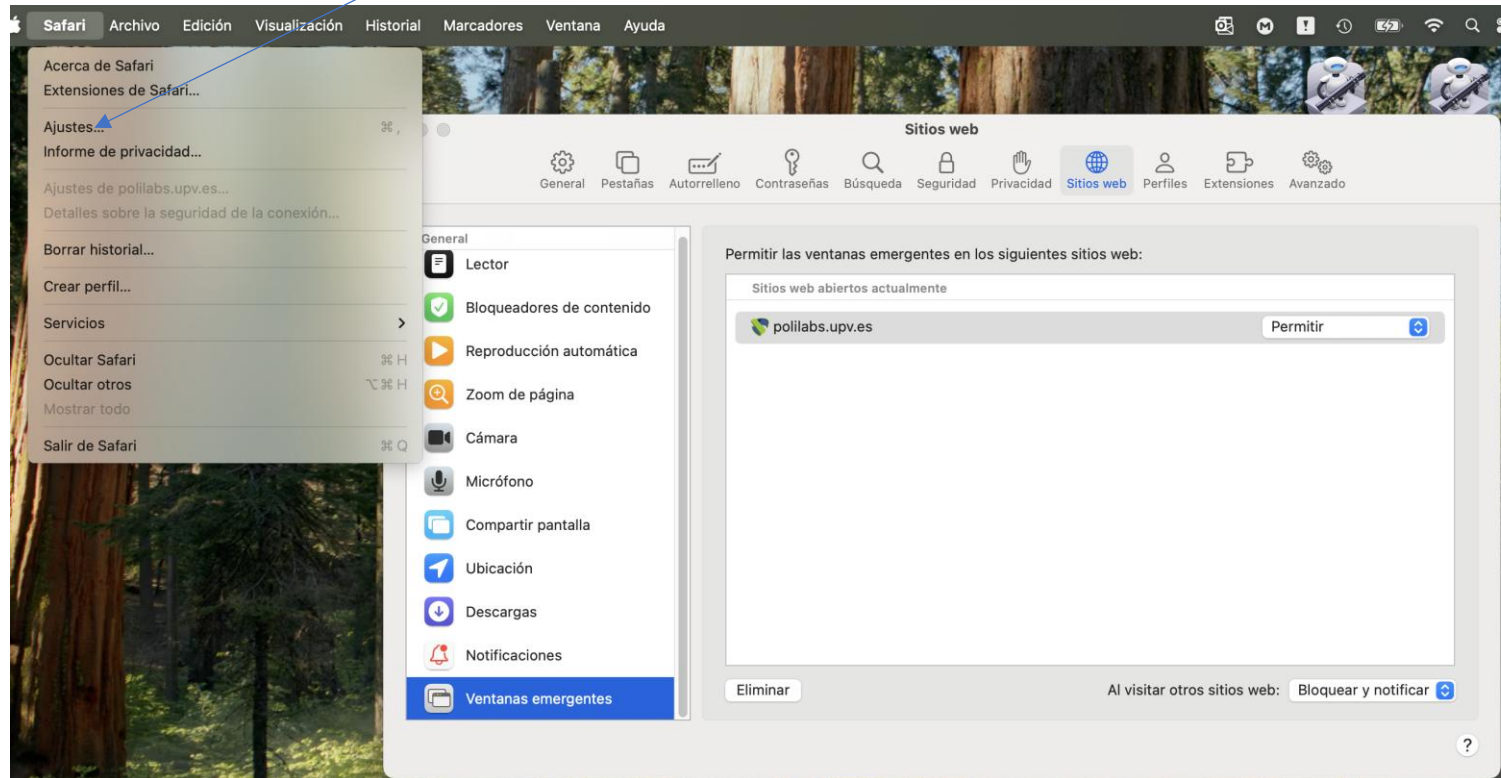
1. Acceso remoto desde Mac

- Para acceder al ordenador virtual pincha en la parte superior del icono correspondiente a la ETSII y selecciona HTML5-RDP.

The screenshot shows the poli[Labs] website interface. The header includes the Universitat Politècnica de València logo, the text 'poli[Labs]', and navigation links like 'UDS Client', 'Manual', 'Soporte', and 'Español'. Below the header is a search bar and a 'Filtro' button. The main content area is titled 'Laboratorios' and displays a grid of virtual lab icons. The icons include 'Aula Gráfica 25-26', 'Aula UPV 2025-26', 'Aulas CFP 1', 'DPI', 'DSiC-GPU', 'DSiC-LINUX', 'DSiC-WINDOWS', 'ETSII', 'ETSINF LIN', 'HTML5-RDP', 'Linux Ubuntu', 'PDI 2024-25', and 'Windows EN'. A red circle highlights the 'ETSII' icon, and a blue arrow points from the text above to the 'HTML5-RDP' option within the ETSII icon's dropdown menu. Below the grid are sections for 'Ofimática', 'Autodesk', 'Campus', and 'Software libre'.

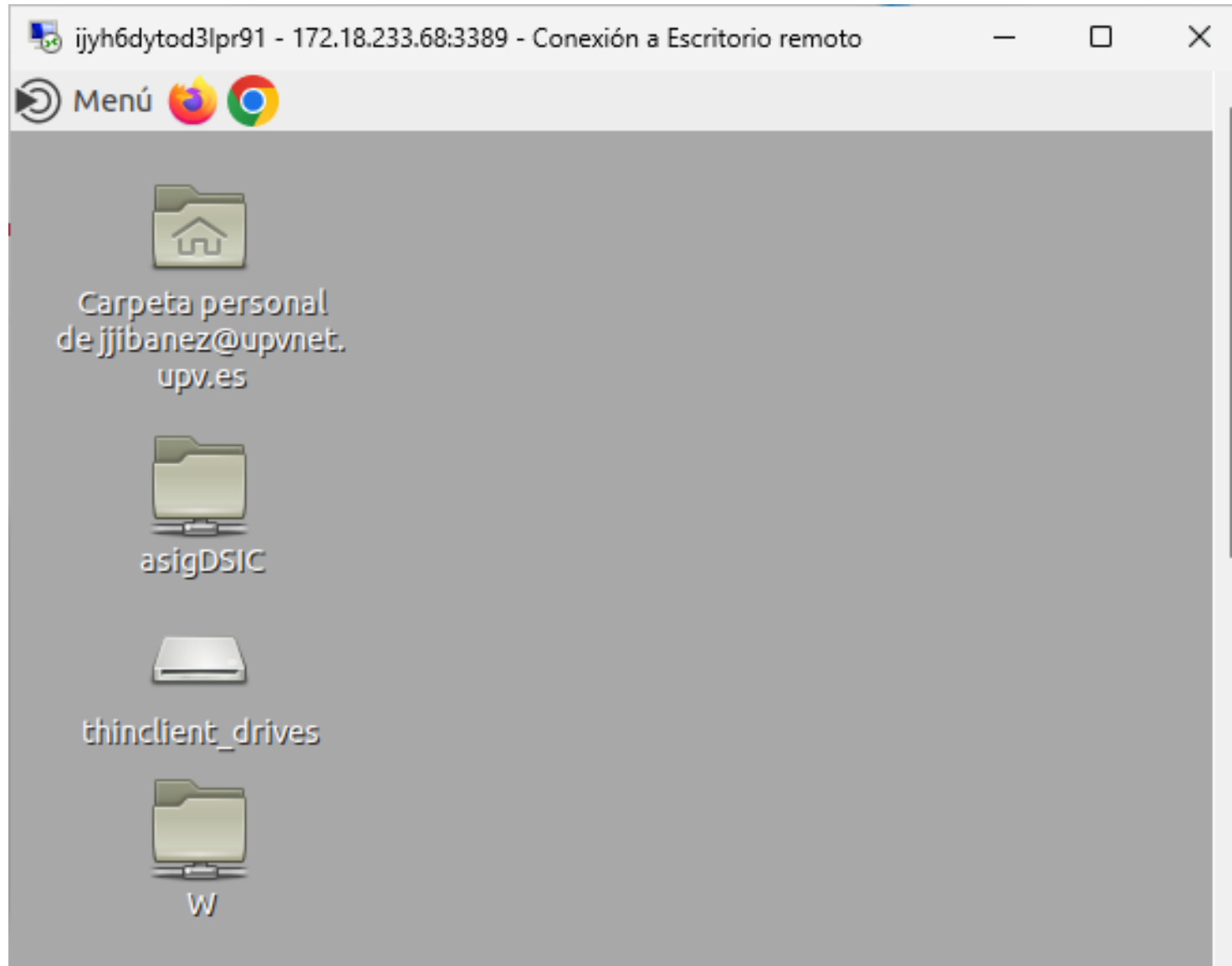
1. Acceso remoto desde Mac

- En el navegador que estés usando, debe estar habilitada la opción de permitir ventanas emergentes. En el caso de **Safari**, la tendrás que habilitar, eligiendo la opción **Ajustes**.



Prácticas en CPA: trabajar con el Escritorio remoto DSIC-LINUX

Una vez te hallas autenticado, aparecerá un escritorio virtual de Linux, similar al que aparece en los ordenadores de prácticas



Prácticas con CPA usando ordenador personal

- **Linux:**
 - Instalar gcc (tiene incorporado OpenMP)
 - Instalar el paquete OpenMPI (memoria distribuida)
- **MAC** (el gcc de MAC no tiene incorporado OpenMP)
 - Instación de gcc y OpenMPI mediante Homebrew (<https://brew.sh/>)
 - Instación de gcc y OpenMPI mediante **MacPorts**



Prácticas con CPA usando el ordenador personal

Windows 10/11: Es posible instalar un terminal Linux de Ubuntu mediante **Microsoft store** (es gratuita) e instalar gcc y el paquete OpenMPI:

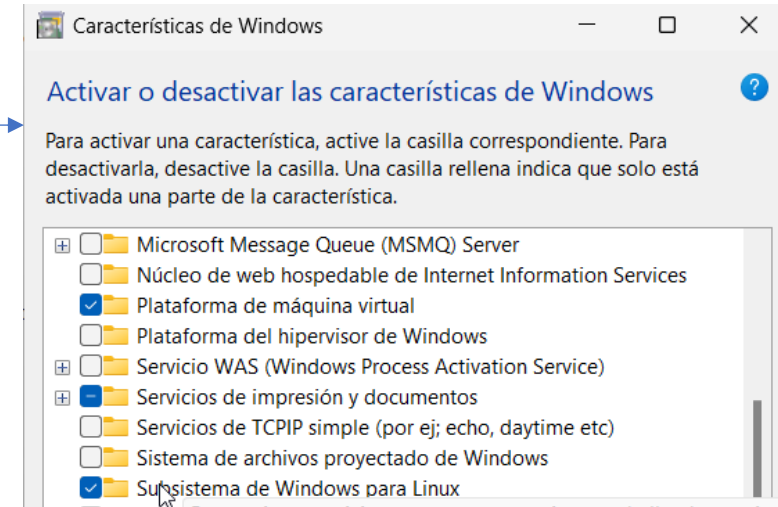
1. Activar **Plataforma de máquina virtual y Subsistema de Windows para Linux**
2. En **Microsoft Store**, descargar una versión de Linux(Ubuntu)
3. Si aparece un mensaje de error al ejecutar la aplicación instalada, debemos actualizar el **Subsistema de Windows para Linux (wsl)**. Para ello, desde el programa **Windows PowerShell**, ejecutaríamos el siguiente comando: `wsl.exe -update`
4. Para instalar gcc (OpenMP incorporado) y OpenMPI, desde el terminal de Ubuntu ejecutaríamos los siguientes comandos:

```
sudo apt-get clean
```

```
sudo apt-get update
```

```
sudo apt install gcc
```

```
sudo apt install libopenmpi-dev
```



Conexión al Cluster Kahan

- Puedes conectarte al Front-end de kahan mediante una **conexión VPN** usando:

```
ssh -l nombre_usuario@kahan.dsic.upv.es
```
- Puedes conectarte también mediante **ssh** con el programa Putty (Windows y Mac OS)
- Una vez os hayáis conectado al front-end, se deben ejecutar los programas en el front-end y lanzar los trabajos al sistema de colas del cluster Kahan
- Los códigos implementados los puedes copiar al Cluster de Kahan, usando los programas:
 - Windows: programa Winscp
 - Linux: comando **scp** desde un terminal Linux
 - Mac: programa CyberDuck o **scp**

hola_openMP.c (código fuente)

```
#include <stdio.h>
#include <omp.h>
int main () {
    int nthreads;
    #pragma omp parallel /*Crea un Región Paralela, activando los hilos*/
    {
        int id = omp_get_thread_num(); // cada uno obtiene su identificador de hilo
        nthreads = omp_get_num_threads(); // todos obtienen el número de hilos
        printf("Hola desde el thread %d de %d threads\n", id, nthreads);
    }
}
```

Ejemplo de ejecución en el Front-end y en el cluster

hola.sh (script para lanzar el trabajo)

```
#SBATCH --nodes=1
#SBATCH --time=5:00
#SBATCH --partition=cpa
./hola
```

```
jjibanez@EVIRL-035-OK:~$ ssh jjibanez@kahan.dsic.upv.es
jjibanez@kahan.dsic.upv.es's password:
Last login: Thu Oct  1 18:42:36 2020 from asgard.dsic.upv.es
tabs: no tab-list given
-bash-4.1$ cd codigos_OpenMP/
-bash-4.1$ ls
aux_mat.c  funciones_basico.c  hola_openMP.c  matrices.c
codigos    hola                hola.sh        omp_for_vector.c
-bash-4.1$ gcc -fopenmp -o hola hola_openMP.c
-bash-4.1$ ./hola
Hola desde el thread 1 de 4 threads
Hola desde el thread 2 de 4 threads
Hola desde el thread 0 de 4 threads
Hola desde el thread 3 de 4 threads
-bash-4.1$ qsub hola.sh
260659
-bash-4.1$ ls
aux_mat.c  funciones_basico.c  hola_openMP.c  hola.sh.e260659  matrices.c
codigos    hola                hola.sh        hola.sh.o260659  omp_for_vector.c
-bash-4.1$ cat hola.sh.o260659
Hola desde el thread 10 de 32 threads
Hola desde el thread 2 de 32 threads
Hola desde el thread 18 de 32 threads
Hola desde el thread 14 de 32 threads
Hola desde el thread 16 de 32 threads
Hola desde el thread 6 de 32 threads
Hola desde el thread 1 de 32 threads
Hola desde el thread 19 de 32 threads
Hola desde el thread 31 de 32 threads
Hola desde el thread 17 de 32 threads
Hola desde el thread 13 de 32 threads
```

→ Conexión ssh al Front-End

→ Compilación código en el Front-End

→ Ejecución del código en el Front-End

→ Ejecutar el código en el cluster: hay que poner en la cola de Kahan el ejecutable (el sistema de colas le asigna un número)

→ Por defecto, genera dos ficheros: uno para ver los errores y otro para ver las salidas obtenidas al ejecutar el código en el cluster