

Arquitectura de Computadoras Armado de Ambiente de Trabajo

Introducción	1
GNU/Linux instalado en forma nativa.....	1
Workspace del ITBA	3
Puedo trabajar con archivos en mi disco local?	7
Máquina virtual.....	8
Windows	10
Prerrequisitos	11
¿Cómo activo el subsistema?.....	11
¿Cómo correr aplicaciones gráficas?	13
Trucos adicionales y sugerencias.....	15
Para aquellos con MAC que usa chip ARM.....	16

Introducción

Las prácticas están desarrolladas para usar Linux Ubuntu 64 bits con algunos programas adicionales a instalar.

Durante la primera mitad del cuatrimestre podrán realizar las prácticas usando el Workspace de ITBA sin necesidad de instalar nada adicional en sus máquinas.

A aquellos que ya tienen **Ubuntu en forma nativa** o virtualizada y quieren seguir usando ese ambiente, revisen si además disponen de los siguientes programas que van a necesitar. Si no los tienen, pueden instalarlos:

- Bless (Hexadecimal editor). Como alternativa también puede ser el Gnome Hexadecimal Editor (Ghex).
- EDB (Evan's Debugger)
- Docker
- Nasm
- QEMU

La otra alternativa es usar una máquina virtual Ubuntu que ya tiene todo instalado.

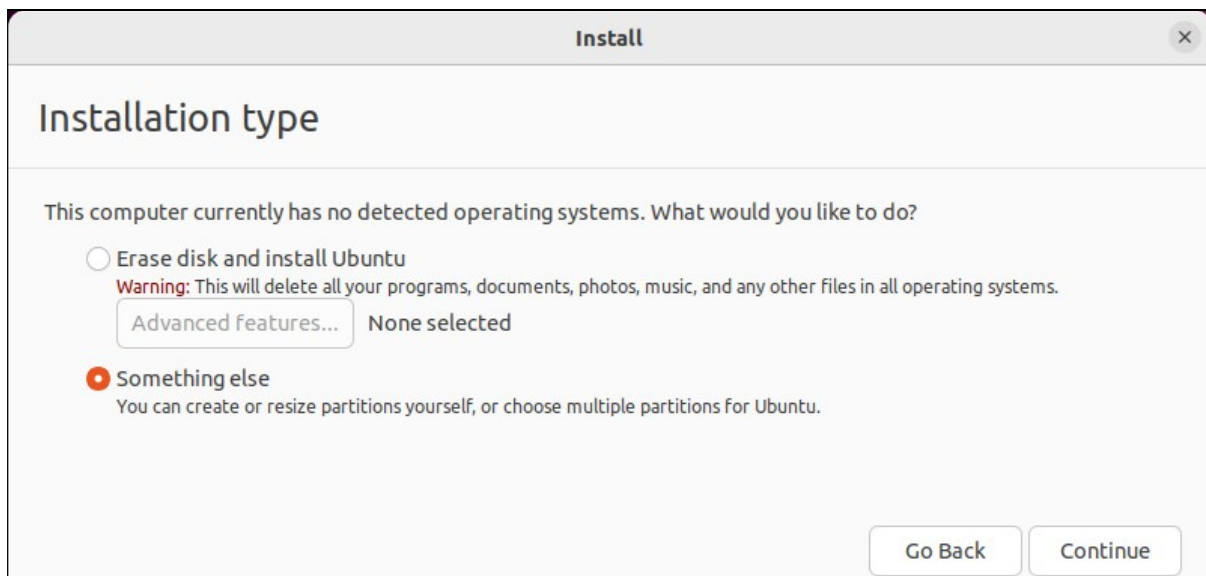
A continuación se describe cada una de las opciones que pueden elegir para el entorno de desarrollo a utilizar.

GNU/Linux instalado en forma nativa

Se pueden descargar una distribución de Linux (Ubuntu, por ejemplo) e instalarla directamente en la máquina. El procedimiento es simple:

- Descargan la distribución elegida.
- Generan un USB booteable.
- Instalan.

Respecto de la instalación, tienen **2 opciones**: pueden borrar completamente el disco y tener GNU/Linux como único sistema operativo, pero también tienen la opción de mantener Windows (suponiendo que ya lo tienen instalado en sus máquinas) y generar una nueva partición donde va a residir el nuevo sistema operativo (es una de las opciones seleccionables durante la instalación de Ubuntu).



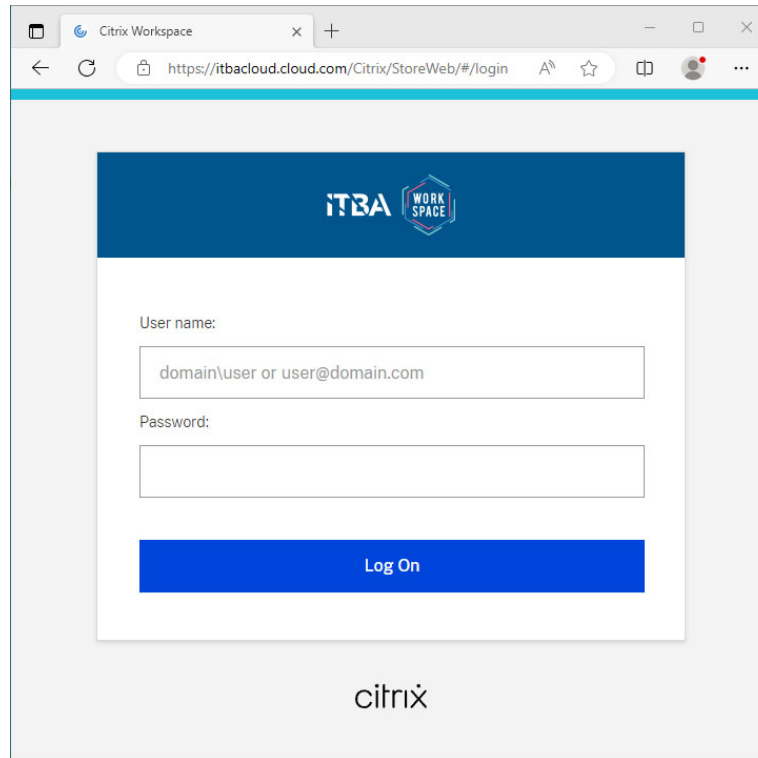
Luego, al iniciar la máquina, podrán elegir cuál de ellos quieren bootear.

Workspace del ITBA

El siguiente ejemplo ha sido realizado bajo Windows 10 Pro¹ e instalando la versión de Citrix Workspace 2307.

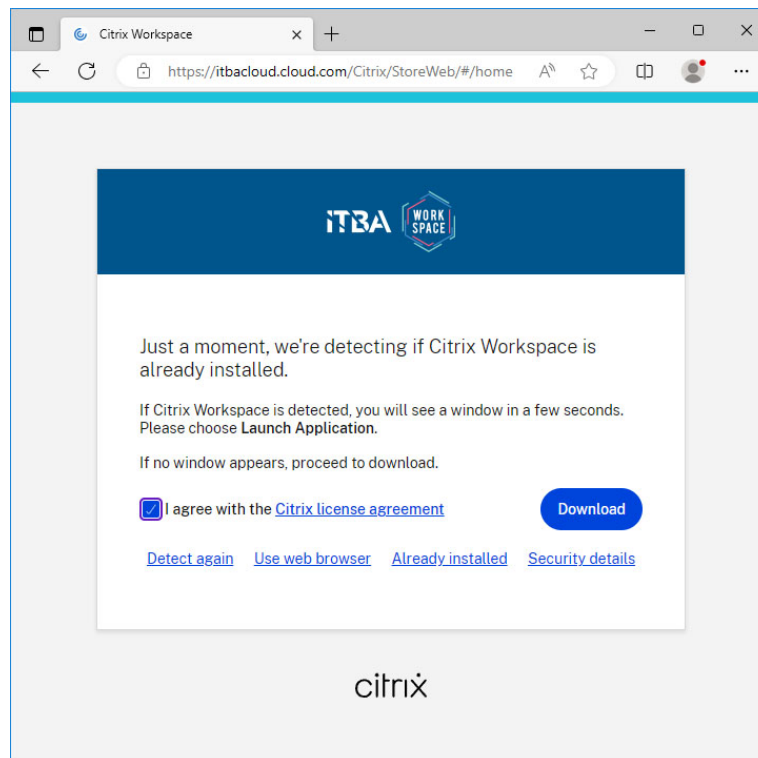
<https://workspace.itba.edu.ar/>

1.

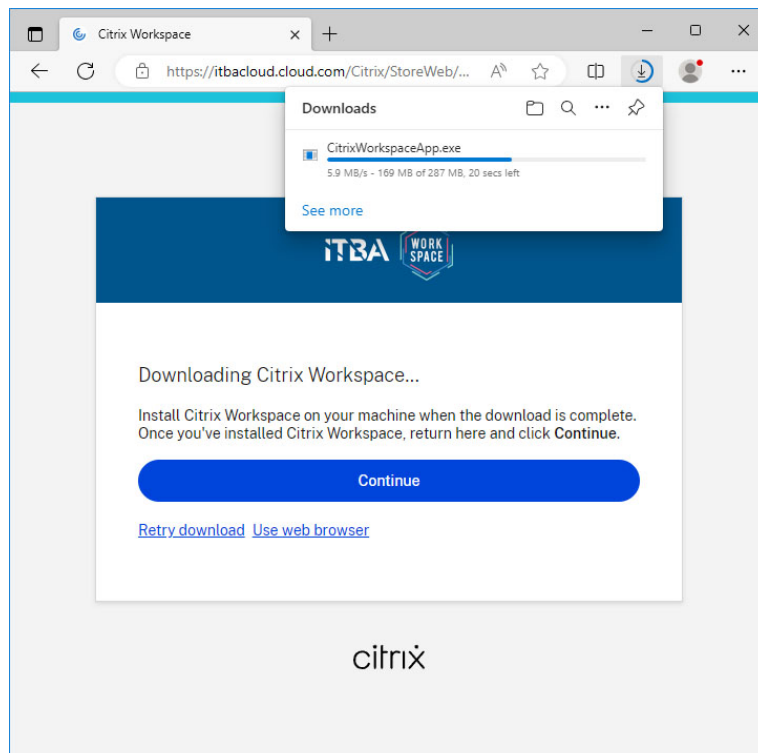


¹ Versión 1607, Build 14393.2189

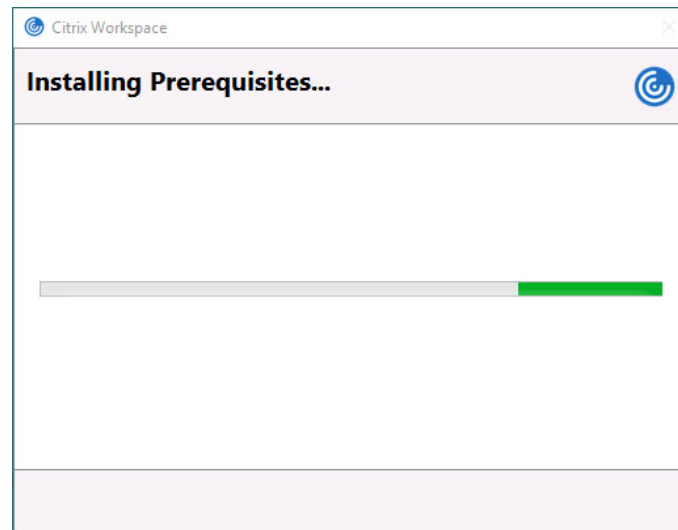
2.



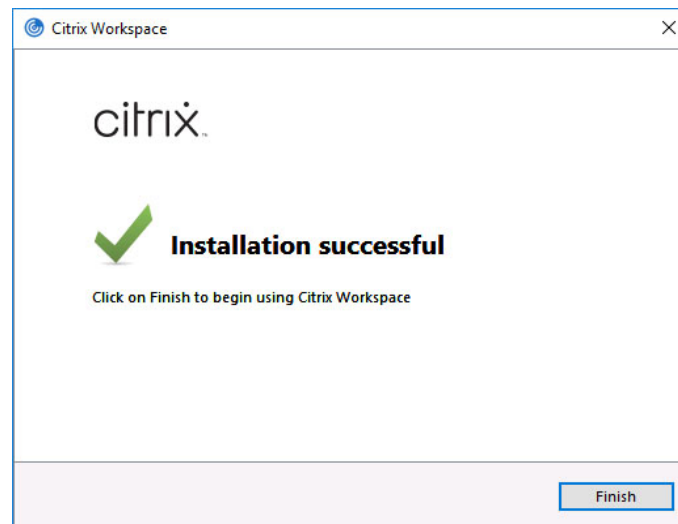
3.



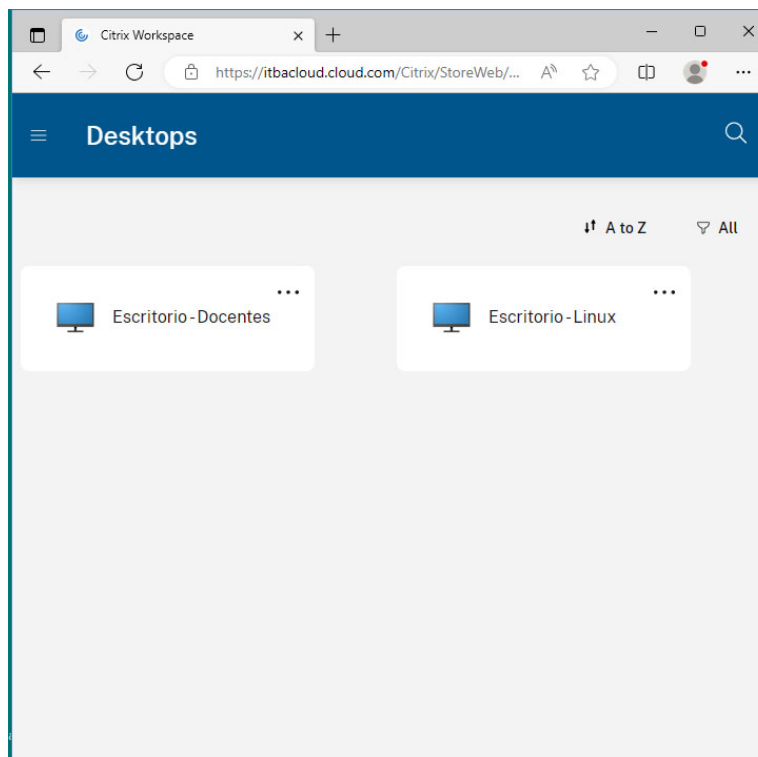
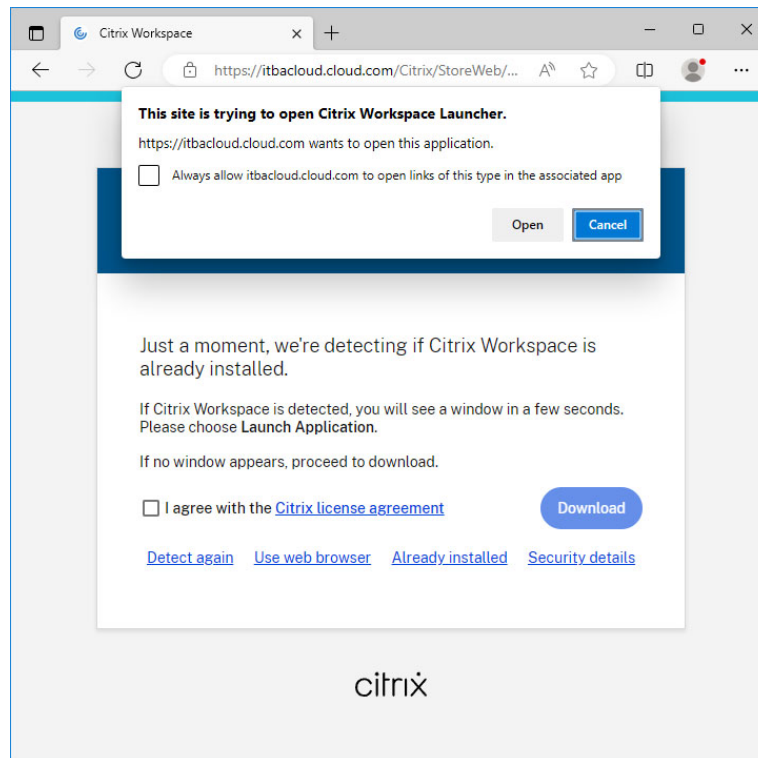
4.



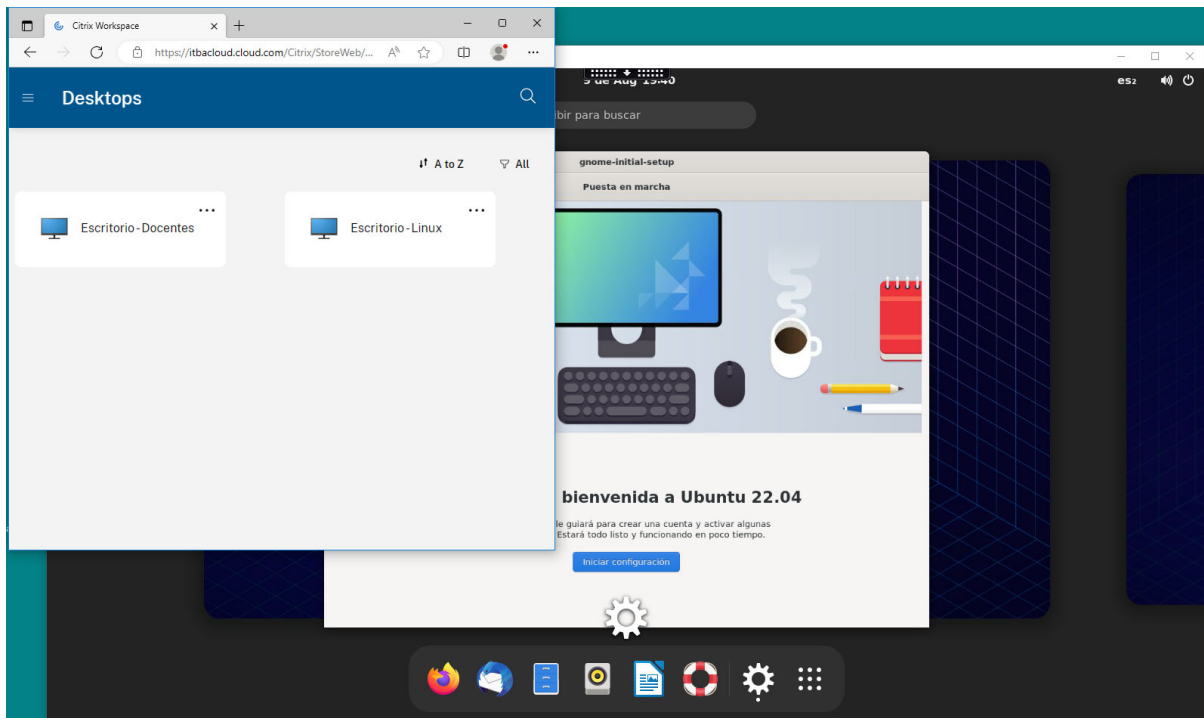
5.



6. Reinicie la computadora.
7. Volver a loguearse en <https://workspace.itba.edu.ar/>
8. Permitir que abra el Workspace.

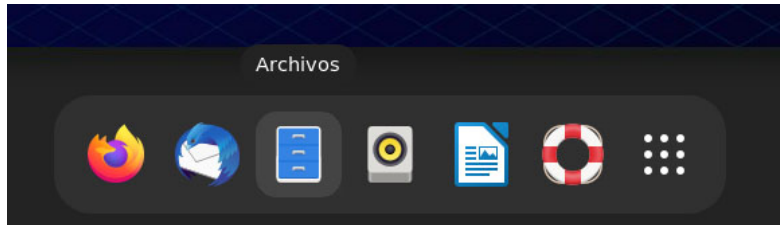


9. Ya tenemos el escritorio Linux Ubuntu 22.04



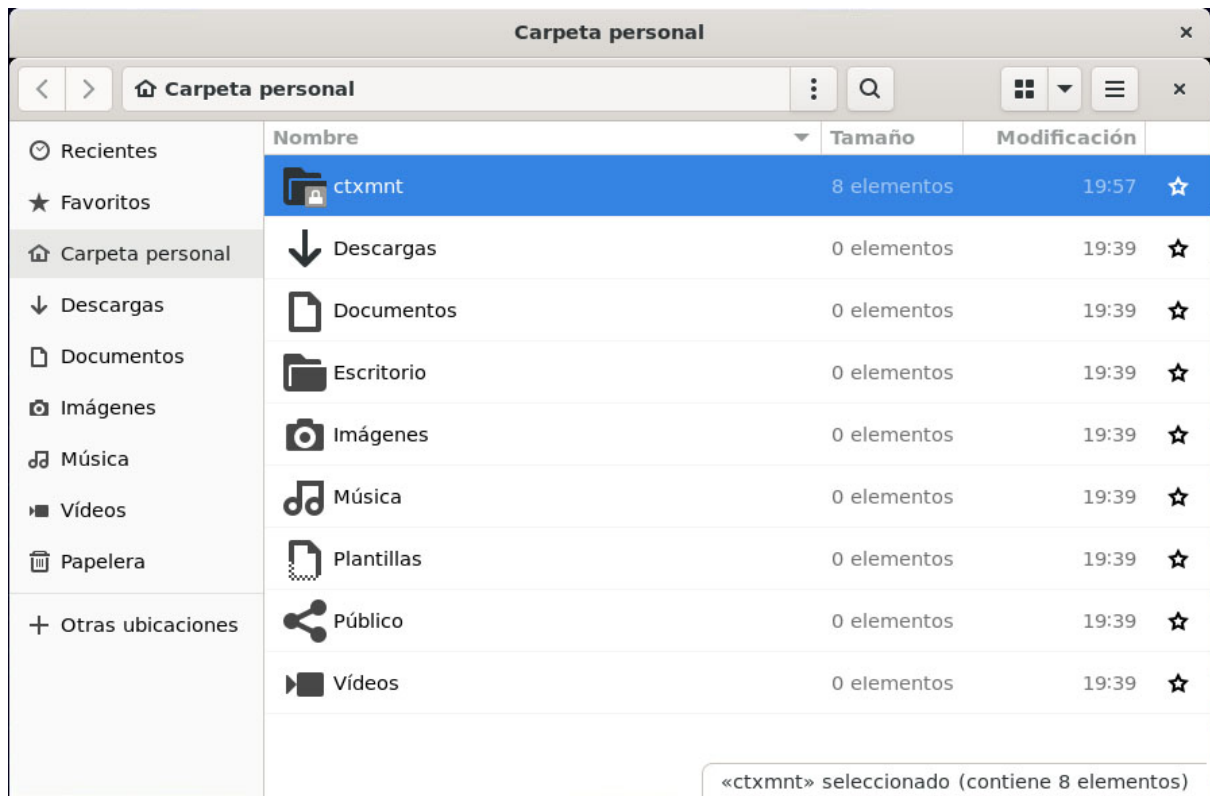
Puedo trabajar con archivos en mi disco local?

Abrir “Archivos”



Les solicitará autorización para acceder a sus archivos locales.

Verán sus discos montados en el sistema de archivos.



Máquina virtual²

La misma esta exportada desde Virtualbox y zipeada en:

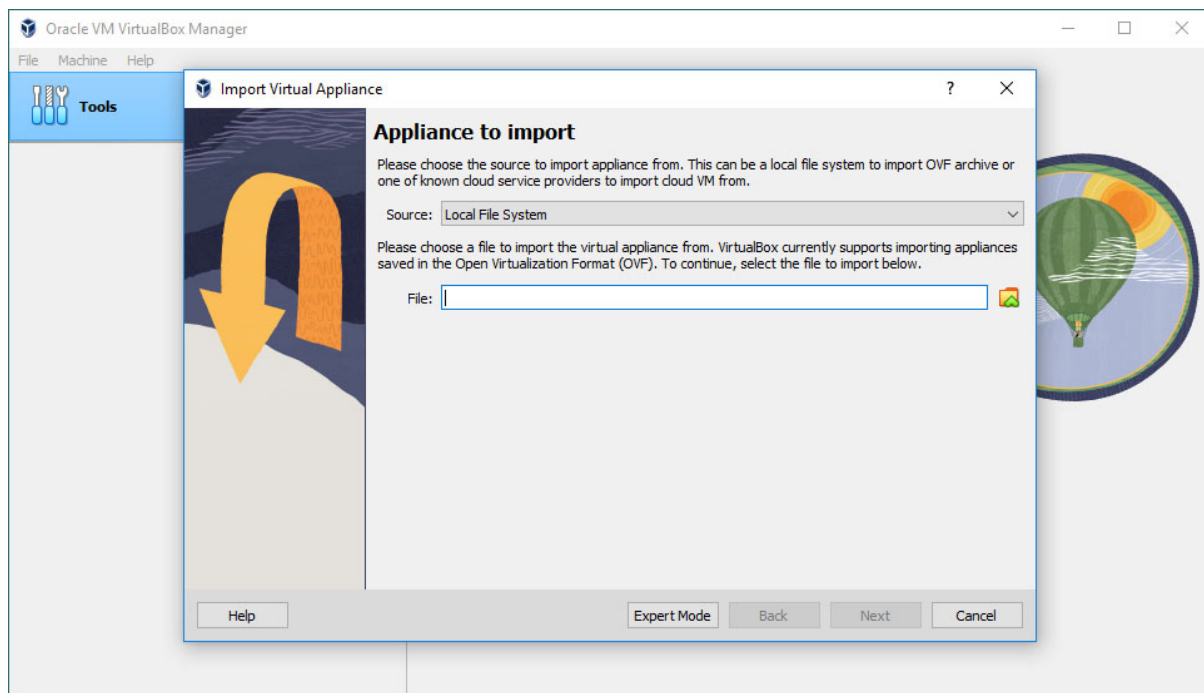
https://drive.google.com/file/d/1zoTyMFTj4R0m3s22bcmvXZdoyzXNHx9M/view?usp=drive_link

- Usuario = arq
- Password = arq

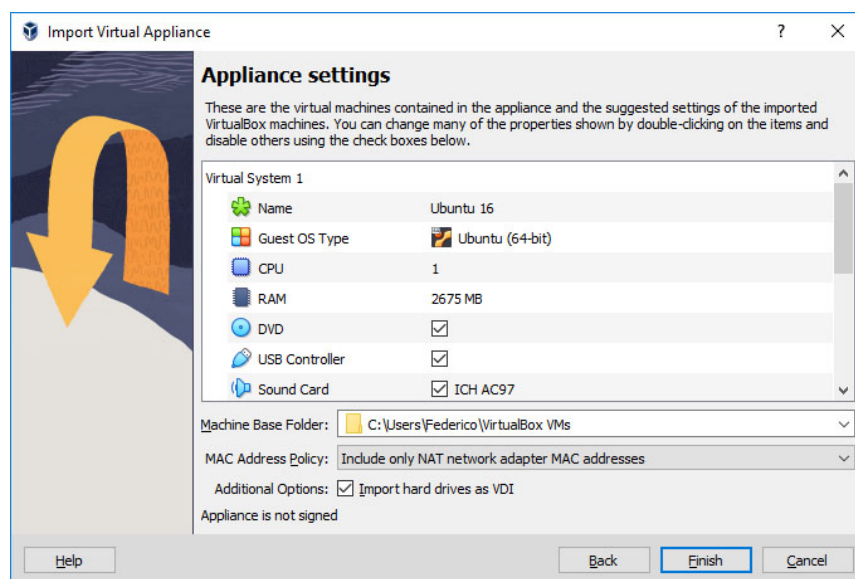
Importar la máquina virtual:

² Las capturas de pantalla mostradas corresponden a VirtualBox Version 7.0.10 r158379 (Qt5.15.2)

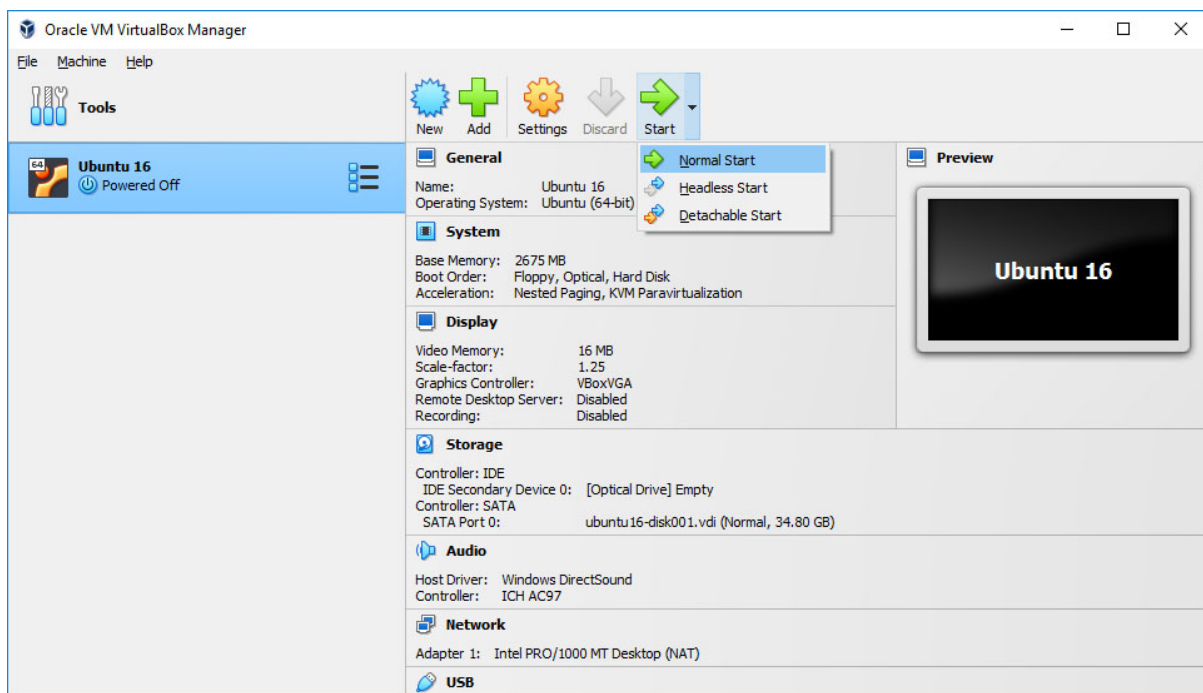
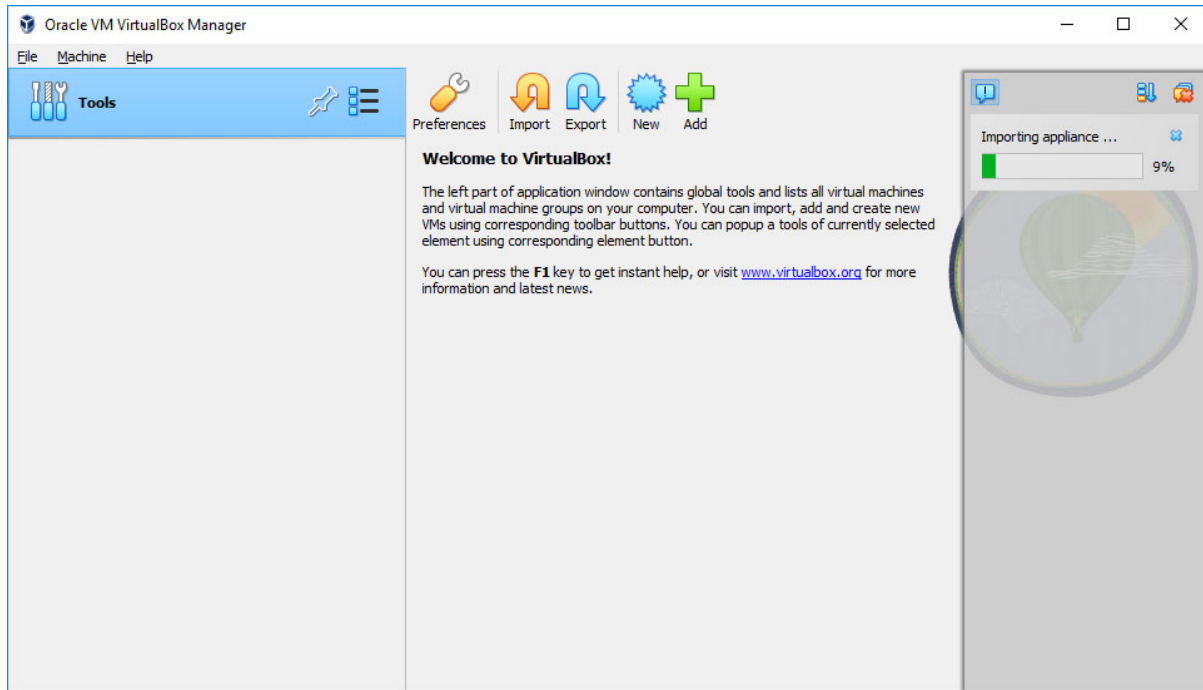
- VirtualBox → Import



- Busque el archivo .ova.



- Le mostrará la configuración del dispositivo. Finalice la importación.
- Listo, una vez que termine, podrá arrancar y usar la máquina.



Windows

Para aquellos que usan Windows explicaremos cómo configurar Windows 10 para poder lidiar con el trabajo que contempla la materia sin recurrir a una máquina virtual.

Debido a que las utilidades sugeridas por la cátedra corren sobre Linux, utilizaremos una funcionalidad de Windows 10 llamada Windows Subsystem for Linux para poder correrlas nativamente en Windows.

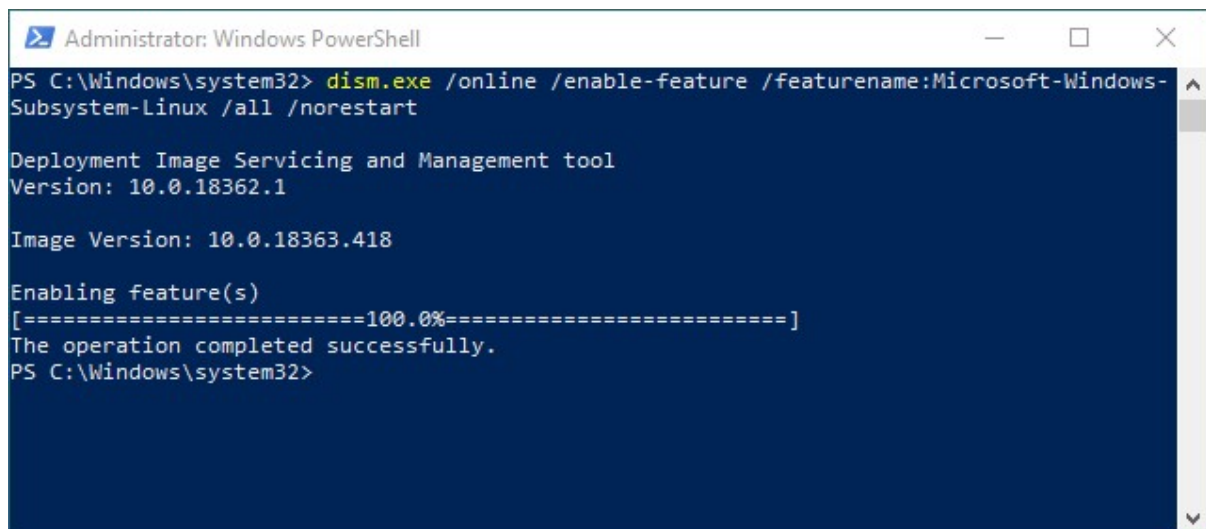
Prerrequisitos

Si bien Windows 10 incluye la primera versión del Subsistema Linux desde 2016, la actualización de mayo 2020 introduce una nueva versión, llamada WS2. Esta actualización lleva la compatibilidad de system calls al 100% y mejora abismalmente la performance del sistema de archivos. WSL2 requiere la actualización de mayo (Windows 10 versión 2004, compilación 19041 o superior), y recomendamos su uso por sobre la versión 1 en especial de cara al trabajo final. Por favor, actualizar Windows 10 a la última versión disponible antes de continuar.

¿Cómo activo el subsistema?

El primer paso es abrir una terminal PowerShell como administrador y ejecutar los siguientes comandos:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

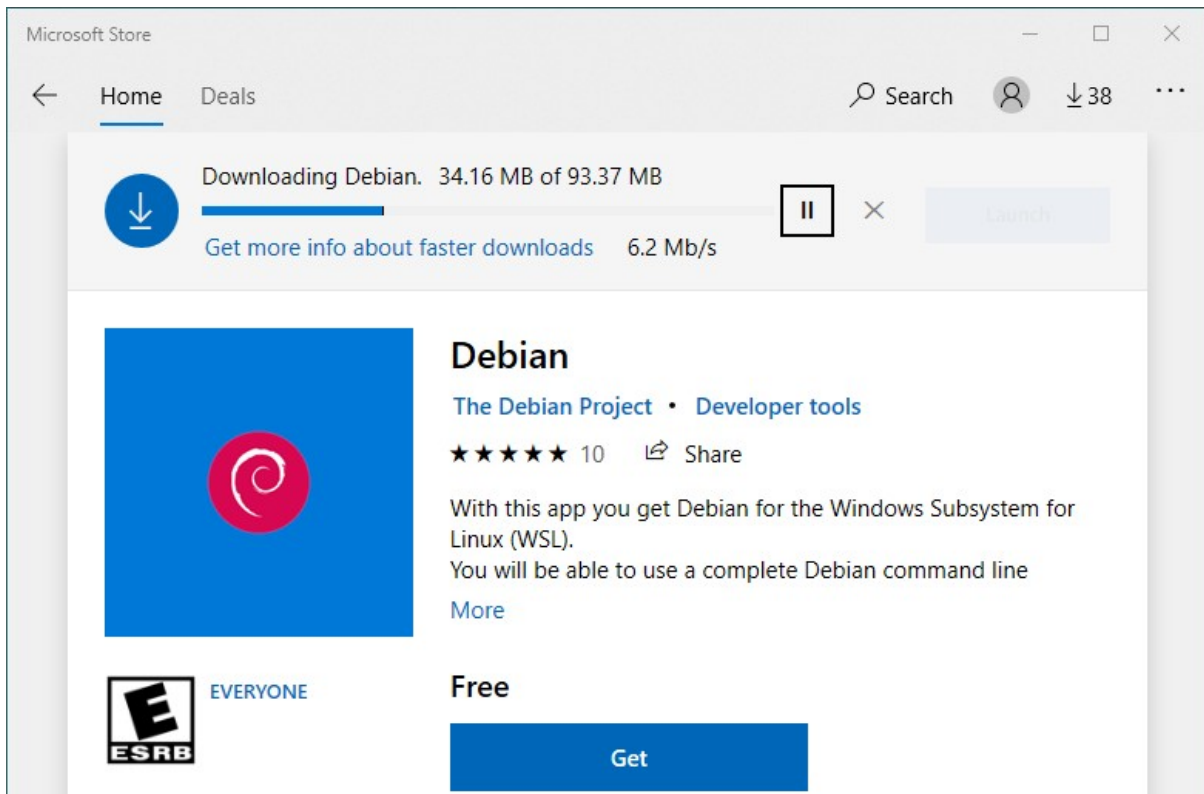


Reiniciar la computadora para aplicar los cambios, y correr, finalmente, el siguiente comando, una vez más en una ventana de Powershell en modo administrador:

```
wsl --set-default-version 2
```

Los primeros dos comandos activan el Subsistema Linux y la Plataforma de Máquina Virtual de Hyper-V, necesaria para correr WSL2. El tercer comando establece que WSL2 será la versión por defecto para distribuciones que se instalen desde la Tienda Windows a partir de ese momento. Es posible alternar fácilmente entre WSL1 y WSL2. La información acerca de cómo lograr eso se encuentra disponible en el sitio web de Microsoft.

A continuación, abran la Tienda Windows (o Microsoft Store, según la versión de Windows 10 que usen), y descarguen la aplicación Ubuntu o Debian. Si bien son las que personalmente recomiendo, otras distribuciones populares como Suse, Arch y Fedora se encuentran disponibles para su descarga. Cualquiera sea la distribución seleccionada, los pasos siguientes no cambian.



Al abrir la distribución instalada, deberán darle un momento al subsistema para configurarse, y, a continuación, deberán introducir un nombre de usuario UNIX y una contraseña. La misma se utilizará cada vez que quieran correr un comando como superusuario.

```
fgr@DESKTOP-PG39478: ~
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: fgr
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
fgr@DESKTOP-PG39478:~$
```

¡Ya está! Tienen ante ustedes una Shell Linux completamente funcional y compatible con la enorme mayoría de los binarios de Linux. Para terminar, y como harían en cualquier instalación fresca de Debian, para actualizar los paquetes instalados, corran:

```
sudo apt update && sudo apt upgrade
```

La principal limitación del Subsistema radica en su incapacidad para correr aplicaciones con interfaz gráfica. Si bien una actualización del Subsistema con soporte nativo para

aplicaciones gráficas fue ya anunciada, a continuación, los guiamos en el proceso de sortear esta limitación para poder, efectivamente, correr aplicaciones gráficas, como Bless o QEMU.

¿Cómo correr aplicaciones gráficas?

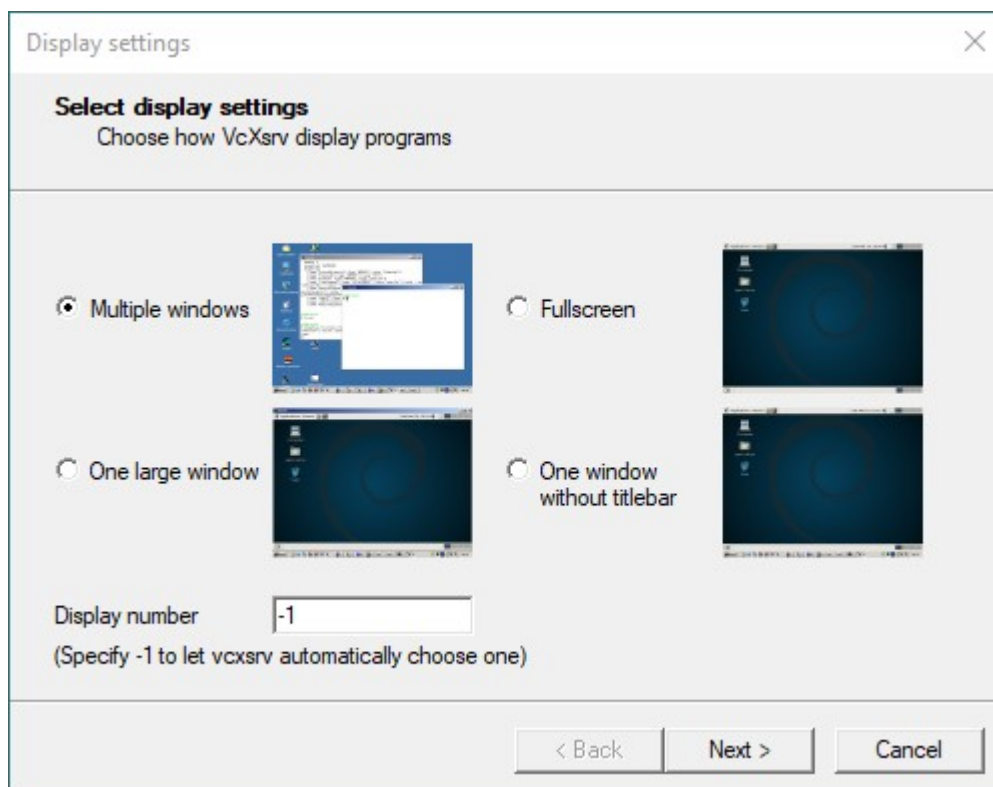
Una de las maneras mediante las cuales Linux provee soporte para aplicaciones gráficas es implementando el protocolo X. Se trata de una serie de APIs a través de las cuales los programas que deseen dibujar en la pantalla pueden comunicarse con un programa llamado XServer, que a su vez es el encargado de administrar la pantalla en sí y operar sobre la misma. Lo que debemos hacer, entonces, es instalar un XServer e indicarle al Subsistema Linux donde encontrarlo.

Uno de los XServer gratuitos más populares para Windows se llama VcXsrv, y es el que utilizaremos en este ejemplo. Se encuentra disponible para su descarga a través del siguiente enlace

<https://sourceforge.net/projects/vcxsrv/>

Descárguenlo e instálenlo.

Cuando lo ejecuten, configúrenlo según se muestra en las siguientes imágenes:



Client startup

Select how to start clients

☒ Start no client
This will just start the xserver. You will be able to start local clients later.

☐ Start a program
This will start a local or remote program which will connect to the xserver. You will be able to start local clients later too. Remote programs are started using SSH.

☐ Open session via XDMCP
This will start a remote XDMCP session. Starting local clients later is limited. This option is not available with the "Multiple windows" mode.

< Back Next > Cancel

Extra settings

Extra settings

☒ Clipboard
Start the integrated clipboard manager
☒ Primary Selection
Also map the PRIMARY selection to the windows clipboard.

☒ Native opengl
Use the native windows opengl library (wgl). Make sure to export the LIBGL_ALWAYS_INDIRECT environment variable.

☒ Disable access control
Use this when you want vcxsrv to accept connections from all clients.

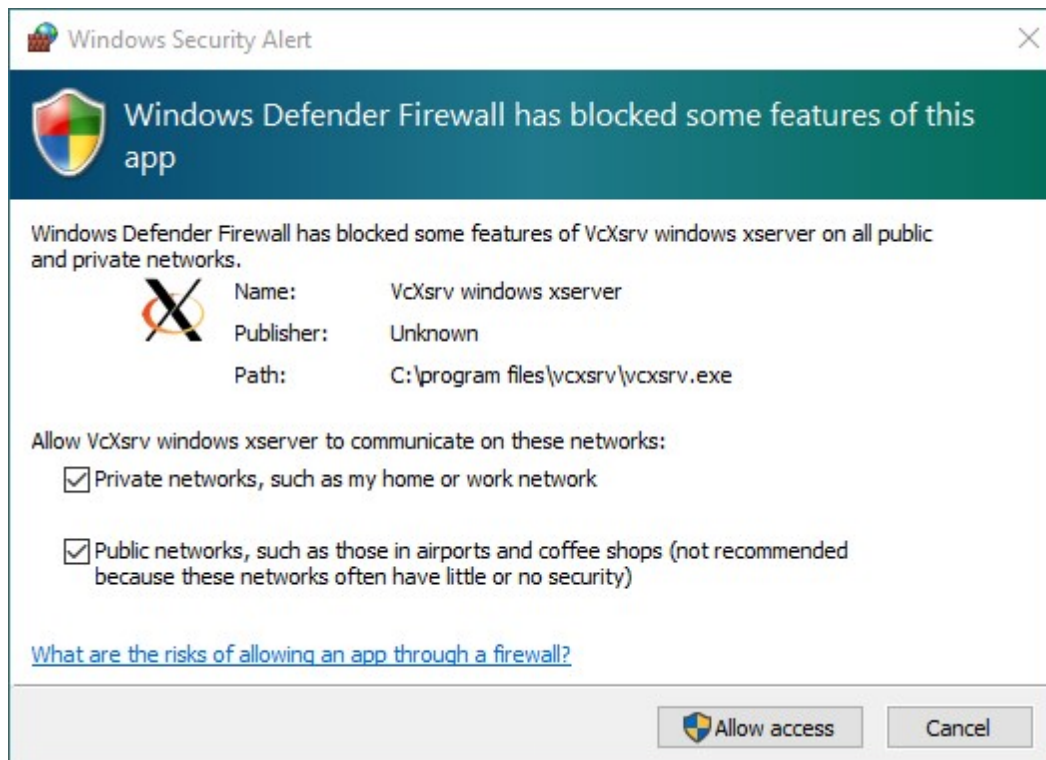
Additional parameters for VcXsrv

< Back Next > Cancel

Esto es necesario debido a que WSL2 maneja su propio adaptador de red, por lo que debe exponerse el XServer a esta interfaz.

Cuando el Firewall de Windows les pida permiso para que el XServer acceda a las interfaces de red, deberán concederle acceso a las redes públicas, puesto que WSL2 se expone a través de una. Tilden la casilla correspondiente a redes públicas, que viene destildada por defecto en la ventana de autorización. Para aquellos preocupados por

cuestiones de seguridad, pueden optar por seguir esta pequeña guía (<https://github.com/cascadium/wsl-windows-toolbar-launcher#firewall-rules>).



El último paso debe realizarse desde WSL2. Ejecuten el siguiente comando:

```
sudo nano ~/.bashrc
```

y utilizando el editor de texto, agreguen, al final del archivo, que acaba de abrirse, las siguientes líneas:

```
export DISPLAY=$(awk '/nameserver / {print $2; exit}' /etc/resolv.conf
2>/dev/null):0
export LIBGL_ALWAYS_INDIRECT=1
```

Presionen CTRL + O para guardar los cambios, y cierren todas las instancias de WSL2. Los cambios estarán disponibles desde la próxima vez que abran WSL2, puesto que las líneas que agregaron se ejecutarán cada vez que una nueva instancia de la Shell se abra. Los comandos agregados indican a WSL2 donde encontrar el XServer que instalamos antes para que pueda comunicarse correctamente con el programa.

¡Listo! Como prueba de concepto, pueden instalar el paquete x11-apps y correr, por ejemplo, xeyes.

¡Recuerden que el XServer debe estar corriendo para poder ejecutar aplicaciones gráficas! Pueden configurarlo para que se ejecute automáticamente al iniciar sesión (hay varios tutoriales sobre esto en Internet) o abrirlo cada vez.

Trucos adicionales y sugerencias

Pueden instalar Windows Terminal desde la Tienda Windows para una mejor experiencia de terminal que la que el venerable símbolo del sistema ofrece.

WSL2 emula un sistema de archivos igual al de Linux. La performance es ideal cuando trabajan en este entorno, por lo que recomendamos que ubiquen todos sus archivos Linux allí. Pueden recorrer el sistema de archivos Linux utilizando el clásico Explorador de Windows escribiendo “explorer.exe” desde cualquier directorio.

WSL2 monta las unidades de disco rígido del equipo bajo la ruta “/mnt/”. Pueden de esa forma acceder a todos sus archivos Windows e incluso crear accesos directos mediante el comando ln (si quisiesen, por ejemplo, añadir un acceso directo a su carpeta personal de Windows en el home de su usuario de Linux). Les recordamos que trabajar desde WSL2 en los directorios de Windows viene acompañado de una importante penalización de performance, por lo que traten de guardar en el filesystem de WSL2 los archivos que pretenden de acceder desde allí. Como dijo un hombre sabio alguna vez, “A WSL2 lo que es de WSL2, a Windows lo que es de Windows”.

Visual Studio Code se integra por defecto con WSL2. Para abrir un proyecto ubicado en el sistema de archivos de WSL2 desde Visual Studio Code, ubíquense en la carpeta del proyecto desde la terminal, y ejecuten “code”.

Existe una distribución de WSL2, disponible en la Tienda Windows, llamada Penguin. La misma está especialmente adaptada para funcionar de manera completa (incluyendo aplicaciones gráficas) sin requerir configuración, además de ofrecer otras tantas facilidades. Si bien es paga, ofrece una alternativa a tener que seguir este tutorial. También se puede obtener de manera gratuita compilándola desde su Github, puesto que es de código abierto.

Esperamos esta guía los ayude a resolver cualquier problema que puedan encontrar.

Para aquellos con MAC que usa chip ARM

Deben instalar el programa UTM, el cual les permitirá armar una maquina virtual (o usar una ya armada) con sistema operativo Linux (se debe elegir específicamente uno para arquitectura AMD64).

UTM emulará el procesador Intel, y de esa manera se podrá trabajar sin problema.

Una VM ya creada la pueden descargar desde el siguiente link:

https://drive.google.com/file/d/180YjlQW6ZbyN4GcrAAGgKODcnoHb_0Qi/view?usp=sharing

- Usuario = santiago
- Password = 1234

Nueva VM con UTM

Si en lugar de usar la arriba mencionada VM, quieren armar desde cero una ustedes, les recomiendo lo hagan con distribuciones de linux que sean **muy** livianas. Buenas alternativas son: Mint o Xubuntu.

- De las varias alternativas de imágenes que les ofrecerán, deben seleccionar aquella imagen para AMD64.
- Luego de arrancar VM, o si durante la instalación del Sistema Operativo deban indicar si quieren virtualizar o emular, deben seleccionar "emulación".
- Luego de terminar instalación, apagar VM.
- Remover la imagen que había seleccionado como "media".
- Al bootear la VM ya tienen a Linux corriendo sobre procesador X86 emulado.