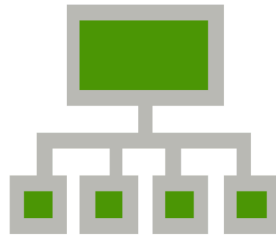


# MultiVNC



Autores: Lorenzo Ferraces, Tomas Gonzalez

Profesor: José Luis Di Biase

## ¿Qué es MultiVNC?

MultiVNC es una aplicación de software libre, multiplataforma basada en la librería LibVNCClient. Esta aplicación permite la conexión a múltiples servidores vnc además de contar con su versión para sistemas Mac OS, Unix, Windows y Android.

## ¿Qué es VNC?

VNC por su lado es un conjunto de programas que permiten la conexión remota y gráfica entre 2 máquinas, usando el protocolo RFB.

## Idea del proyecto

La idea del proyecto es crear un servidor VNC desde una de las máquinas y realizar múltiples conexiones en simultáneo desde distintos dispositivos gracias al MultiVNC.

## Instalación

Para poder utilizar correctamente este programa es necesario contar con las siguientes dependencias.

- cmake
- wxWidgets dev package version >= 3.0
- zlib dev package
- libjpeg dev package

En nuestro caso, tuvimos que instalar otras dependencias más, por errores que aparecían al momento de intentar buildear la aplicación.

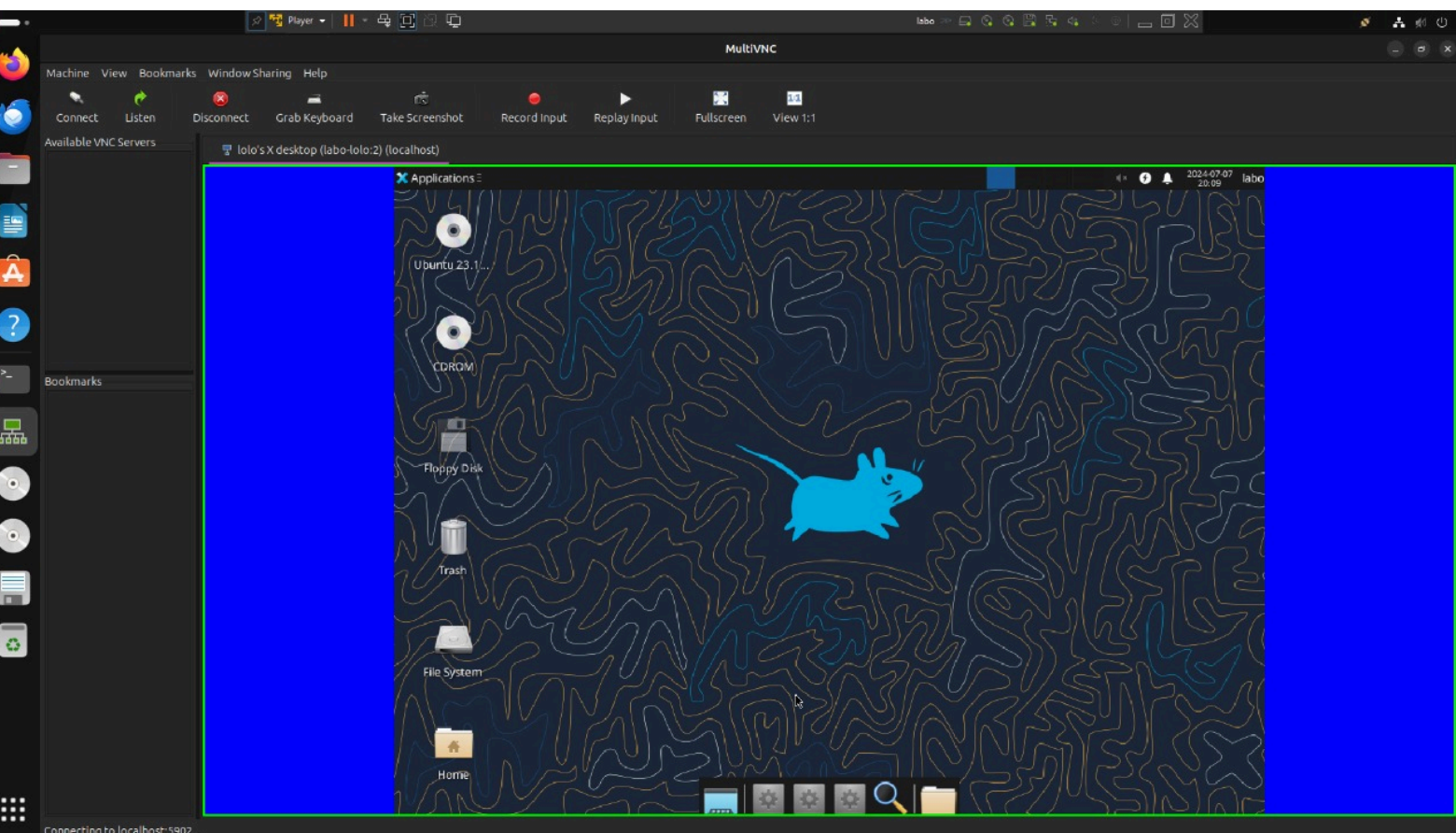
- libwxgtk3.2-dev
- g++
- pkg-config
- libsdl2-dev
- libgtk2.0-dev
- gtk+-3.0

En resumen el código para instalar todas las dependencias es:

```
sudo apt install cmake \  
libghc-zlib-dev lib \  
libjpeg-dev \  
g++ \  
pkg-config \  
libsdl2-dev \  
libgtk2.0-dev  
cmake
```

```
sudo apt-get install gtk+-3.0
```

Una vez que tengamos las principales dependencias instaladas en la máquina, tenemos que clonar el repositorio del proyecto en nuestra maquina local y buildear la aplicación para poder tener el instalador (Obviamente todo esto si queremos nuestra app en un sistema que no sea android).



```
git clone https://github.com/bk138/multivnc  
cd multivnc  
mkdir build; cd build  
cmake ..  
cmake --build .  
cpack  
sudo dpkg -i multivnc_*version*_amd64.deb
```

Por último, para que la conexión sea posible hay que asegurarse que los puertos correctos estén abiertos (5900 para vnc, 5500 para vnc inverso, y 22 para ssh).

```
sudo ufw allow 5900  
sudo ufw allow 5500  
sudo ufw allow 22
```

Una vez abierto en la app, podemos encontrar la opción de conectarnos a un servidor vnc. Si dicha conexión funciona correctamente, entonces se nos abre una interfaz con el contenido del dispositivo al que estamos conectado.

Por último, en caso de querer hacer uso de la funcionalidad de vnc inverso en el cliente de linux, se necesita de un servidor vnc que ofrezca esta funcionalidad tal como x11vnc. Para hacer la conexión de esta manera se deben correr los siguientes comandos con el cliente MultiVNC en modo de escucha:

```
sudo apt install x11vnc  
x11vnc -connect HOST:PUERTO
```

## Problemas Técnicos

A lo largo de la configuración del servidor, nos encontramos con múltiples problemas para la correcta conexión, tales como:

- Errores visuales después de la conexión (pantalla gris)

### Pantalla gris

Si después de conectarnos a un servidor VNC vemos únicamente una pantalla gris, seguramente nos este faltando realizar una configuración en nuestro servidor. En nuestro caso, lo pudimos arreglar reemplazando la configuración inicial del servidor VNC por lo siguiente:

```
#!/bin/sh  
unset SESSION_MANAGER  
unset DBUS_SESSION_BUS_ADDRESS  
startxfce4 &  
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup  
[ -r $HOME/.Xresources ] && xrdp $HOME/.Xresources  
xsetroot -solid grey &
```

***vncconfig -iconic &***

Una vez hecho esto, es necesario verificar también si nuestro servidor usa el protocolo wayland o X11 para la comunicación entre el servidor gráfico y las aplicaciones.

***echo \$XDG\_SESSION\_TYPE***

En caso de ser wayland, tenemos que averiguar cuál es nuestro gestor de pantalla

***cat /etc/X11/default-display-manager***

Si el gestor que tenemos configurado es GDM3 tenemos 2 opciones, la primera desactivar wayland manualmente en la configuración del gestor y reiniciar el servicio, la segunda es instalar el gestor lightDM y configurarlo como gestor de pantalla predeterminado.

En caso de seguir con la primera opción, es necesario modificar el contenido del archivo `/etc/gdm3/custom.conf` con y definir o descomentar la línea

***WaylandEnabled=false***

Una vez hecho esto, reiniciar el servicio con *systemctl restart gdm3*.

Si esto no funciona, entonces podemos probar la segunda opción.

Primero verificamos si contamos con el gestor de lightDM con el comando

***systemctl status lightdm***

En caso de no estar instalado, la terminal nos dirá que no pudo encontrar el servicio. Si es así, lo instalamos y cambiamos el gestor de pantalla predeterminado ejecutando

***sudo dpkg-reconfigure gdm3***

Y eligiendo el gestor LightDM. Una vez hecho esto, podemos cerrar sesión en nuestra máquina y antes de volver a iniciar sesión, cambiamos la opción para usar Xorg en vez de Wayland que es la predeterminada.