



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE ÁLVARO OBREGÓN

INFORME TÉCNICO DE RESIDENCIAS PROFESIONALES

AGOSTO-DICIEMBRE 2023

Instituto Tecnológico de Álvaro Obregón

**IMPLEMENTACION DE COMPUTO EN LA NUBE PARA VIRTUALIZACION DE
APLICACIONES**

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES

RESIDENTE ALEJANDRO HERNANDEZ PIÑA

191210034

ASESOR INTERNO

MCC. RUIZ MARTINEZ JORGE MISael

ASESOR EXTERNO

M. RICARDO GUTIERREZ FERIA

Álvaro Obregón, Ciudad de México a 26 de noviembre del 2023

DEDICATORIA

Para mi querida familia,

En este viaje hacia el mundo del Cloud Computing, su apoyo incondicional ha sido mi mayor fortaleza. Cada paso que he dado en este proyecto ha sido impulsado por su respaldo constante y confianza incansable.

Su presencia ha sido mi mayor motivación, sus palabras de aliento mis alas para volar en este mundo tecnológico. Cada desafío que he enfrentado ha sido más llevadero gracias a su respaldo y ánimo constante.

Dedico este logro a ustedes, quienes con su cariño y apoyo han sido el pilar que sostiene mis sueños y aspiraciones. Gracias por estar siempre presentes, por creer en mí y por ser mi fuerza en este camino.

Con gratitud y cariño,

Alejandro Hernández Piña

AGRADECIMIENTOS

Me dirijo con profundo agradecimiento a todos aquellos que han sido pilares fundamentales en el desarrollo y culminación de mi proyecto de residencia profesional en Cloud Computing, específicamente en la virtualización de máquinas virtuales en entornos Windows y Linux.

Agradezco de manera especial a mi asesor interno, Jorge Misael Ruiz Martínez, por su invaluable orientación, apoyo constante y por brindarme acceso a los recursos tecnológicos necesarios, incluyendo los servidores que fueron vitales para llevar a cabo este proyecto.

Asimismo, mi gratitud se extiende a mi asesor externo, Ricardo Gutiérrez Feria, cuya orientación y conocimientos han sido fundamentales para el éxito de este proyecto. Su visión y aportes fueron de gran valor en cada etapa de este proceso.

No puedo dejar pasar mi agradecimiento al profesor Yilbert Pacheco por su apoyo continuo y sus valiosas contribuciones que fueron fundamentales para la conclusión exitosa de este proyecto.

Además, quiero expresar mi reconocimiento y gratitud al Instituto Tecnológico de Álvaro Obregón por brindarme el entorno propicio y los recursos necesarios para llevar a cabo este proyecto académico.

Su confianza, apoyo y sabiduría fueron esenciales en cada paso de este viaje. Este logro no habría sido posible sin el respaldo de cada uno de ustedes.

Con aprecio y gratitud sincera,

Alejandro Hernández Piña

RESUMEN

En el Instituto Tecnológico de Álvaro Obregón, se enfoca en implementar un proyecto que abarca desde la virtualización hasta la administración de sistemas. Utilizando herramientas como QEMU y KVM, se establece un entorno de virtualización robusto. Además, se emplea Nextcloud para ofrecer una nube privada, asegurando el control y colaboración de datos. La metodología DevOps se guía en 5 etapas que son Desarrollo, Prueba, Integración, Despliegue y Monitoreo.

Hemos finalizado con la implementación de un entorno Nextcloud que ofrece acceso al protocolo SPICE. Esto nos permite visualizar gráficamente las máquinas virtualizadas con sistemas operativos tanto Windows como Linux. En este entorno, se encuentran aplicaciones fundamentales para la carrera de TICs, tales como Android Studio y Visual Studio Code. Además, el uso de aplicaciones como FileZilla y PuTTY ha sido crucial en el desarrollo exitoso de este proyecto.

CONTENIDO

DEDICATORIA	2
LISTAS DE TABLAS	8
LISTAS DE FIGURAS.....	9
CAPITULO 1 Tecnologías implementadas en el desarrollo	20
1.1 Introducción	20
1.2 Generalidades del proyecto.....	21
1.3 ¿Qué significa “Cloud Computing”?	21
1.4 Tipos de modelos de implementación de Cloud Computing.....	22
1.4.1 Nube publicas.....	22
1.4.2 Nube privada.....	22
1.4.3 Nube híbrida.....	22
1.4.4 Nube multiclouds	23
1.5 Tipos de servicios de Cloud Computing	25
1.5.1 Infraestructura como servicio (IaaS).....	25
1.5.2 Plataforma como servicio (PaaS).....	25
1.5.3 Software como servicio (SaaS)	25
1.5.4 Función como servicio (FaaS)	26
1.6 Tipos de hipervisor	28



1.6.1 Hipervisor de Tipo 1	29
1.6.2 Hipervisor de Tipo 2	30
1.6.3 Otros Subtipos de Hipervisores	30
1.7 PLANTEAMIENTO DEL PROBLEMA	32
1.8 OBJETIVO GENERAL.....	33
1.9 OBJETIVO ESPECIFICO	33
1.10 JUSTIFICACION	34
1.11 DELIMITACIONES.....	34
CAPITULO 2 MARCO CONCEPTURAL	36
2.1 DEBIAN 12 SERVER	36
2.2 QEMU.....	38
2.3 KVM	40
2.4 APACHE2.....	42
2.5 NEXTCLOUD	43
2.6 Script .sh en Bash.....	44
2.7 BRIDGE.....	45
2.8 .ISO	46
2.9 SERVIDOR Dell PowerEdge R440.....	46
2.10 PuTTY.....	47

2.11 FileZilla.....	48
2.12 Protocolo SPICE	49
CAPITULO 3 METOLOGÍA DevOps	51
3.1 Metodología DevOps	51
3.2 Desarrollo.....	52
3.3 Prueba	58
3.4 Integración	59
3.5 Despliegue.....	68
3.6 Monitoreo	72
Conclusión	75
Recomendaciones	75
Experiencia profesional adquirida.	76
Referencias.....	77
ANEXO A.....	79
ANEXO B	109

LISTAS DE TABLAS

Tabla 1. Empresas en la Adopción de Modelos de Implementación de Cloud Computing	24
Tabla 2 Empresas de Tecnología y sus tipos de Nube	27
Tabla 3. Comparación entre Hipervisores Tipo 1 y Tipo 2	32
Tabla 4 Características Destacadas de Debian 12.....	37
Tabla 5. Características de QEMU	39
Tabla 6. Características de Kernel-based Virtual Machine	41
Tabla 7. Características del protocolo SPICE.....	49
Tabla 8. Requerimientos mínimos de instalación de Debian 12 Server	52
Tabla 9. Requerimientos Mínimos de instalación del Hipervisor QEMU.....	52
Tabla 10. Requerimientos de Hardware y Software de la prueba y error.....	53
Tabla 11. Requerimientos de Hardware y software del servidor principal.....	54
Tabla 12. Comparativa entre Debian y Ubuntu	60

LISTAS DE FIGURAS

Figura 1. (Se muestra cómo se ven los diferentes modelos de cloud computing.) (Stackscale, 2023)	23
Figura 2. “Cada modelo de servicio en la nube cubre diferentes necesidades de usuarios y empresas” (Stackscale, 2023)	26
Figura 3. Comparación de hipervisores de tipo 1 y tipo 2 (Stackscale, 2023).	31
Figura 4. Arquitectura del funcionamiento de QEMU.	39
Figura 5. Arquitectura del funcionamiento de KVM.	40
Figura 6. Arquitectura de Nextcloud	44
Figura 7. Arquitectura de terminal con el kernel	45
Figura 8. Arquitectura del funcionamiento de Bridge con KVM	46
Figura 9. Arquitectura del funcionamiento del protocolo SPICE	50
Figura 10. Ciclo DevOps (datascientest, 2023)	51
Figura 11. Aplicación Rufus en su modo portable.	55
Figura 12. Página oficial de Debian	55
Figura 13. Configuración de Rufus para Bootear la ISO	56
Figura 14. Configuración de la IP para conectar al servidor por SSH	57
Figura 15. Terminal remota del servidor por SSH	57
Figura 16. Remote Viewer con la ip necesaria	59
Figura 17. Ejecución de la maquina con Windows 10 usando SPICE y remote viewer	59
Figura 18. Comando para hacer una copia de seguridad a la base de datos.	62



Figura 19. Comando para importar la base de datos a nuestro nuevo servidor.	62
Figura 20. Se empaqueta la carpeta en un .tar	62
Figura 21. Comprimimos el archivo empaquetado en un .xz con nivel 9 para tener menos peso del archivo	62
Figura 22. Usamos FileZilla para pasar los archivos correspondientes al servidor Debian.	63
Figura 23. Descomprimimos el archivo	63
Figura 24. Inicio de sesión de nextcloud	64
Figura 25. Página de Inicio de Nextcloud	64
Figura 26. Código para ejecutar el entorno virtual por el protocolo SPICE	65
Figura 27. Código en JS para conectar las ip con los puertos adecuados para el protocolo SPICE	65
Figura 28. Código en HTML para el entorno visual	66
Figura 29. Entorno para ejecutar nuestro sistema operativo	66
Figura 30. Ejecución de Word mediante Cloud Computing	67
Figura 31. Ejecución de Visual Code mediante Cloud Computing	67
Figura 32. Ejecución de Android Studio en modo Cloud Computing	68
Figura 33. Servidor conectado para el despliegue	69
Figura 34. Monitoreo del servidor para ver su estabilidad con el despliegue del servicio SaaS.	69
Figura 35. 6 equipos mostrando la virtualización desde nextcloud	70
Figura 36. Se muestra la aplicación visual code y mysql workbench	70
Figura 37. Se muestra la aplicación Android studio y el Windows sin ejecutar nada	71
Figura 38. Se muestra que también se puede ejecutar Linux como Ubuntu	71

Figura 39. Aplicación de monitoreo de procesos y rendimiento llamado TOP	72
Figura 40. Aplicación de monitoreo de procesos y rendimiento llamado HTOP con entorno más grafico	73
Figura 41. Si desconocemos un proceso, debemos asegurarnos de terminarlo para evitar cualquier posible infiltración.	73
Figura 42. Procesos eliminados	74
Figura 43. Programa Rufus ejecutando.....	79
Figura 44. Página Web Oficial para descargar Debian 12.....	79
Figura 45. Aplicación Rufus lista para Bootear Debian en una USB	80
Figura 46. Menú de instalación grafica o en terminal	80
Figura 47. Seleccionamos el idioma del Sistema operativo	81
Figura 48. Seleccionamos nuestro país	81
Figura 49. Seleccionamos nuestro teclado.....	82
Figura 50. Le damos un nombre al servidor	82
Figura 51. Si tenemos un dominio se lo damos, en mi caso no tengo uno asi que lo dejo en blanco	83
Figura 52. Le damos una contraseña en SU.....	83
Figura 53. Le damos un nombre de usuario.....	84
Figura 54. Le damos una contraseña al usuario.....	84
Figura 55. Escogemos la zona horaria, en mi caso es Central	85
Figura 56. Seleccionamos manual para configurarlo a nuestra manera el disco	85
Figura 57. Seleccionamos nuestro disco duro.....	86

Figura 58. Le damos en si	86
Figura 59. Seleccionamos el espacio libre del disco.....	87
Figura 60. Se crea partición nueva.....	87
Figura 61. Seleccionamos el espacio para el SWAP	88
Figura 62. Seleccionamos primaria.....	88
Figura 63. cambiamos utilizar como:.....	89
Figura 64. Seleccionamos el SWAP	89
Figura 65. Aquí ya quedo seleccionado el SWAP	90
Figura 66. Hacemos lo mismo con el espacio que sobra, pero sobre raíz /	90
Figura 67. Con las particiones ya echas procedemos a finalizar el particionado	91
Figura 68. Le damos en si	91
Figura 69. Le damos en si	92
Figura 70. Una vez completado le damos en no	92
Figura 71. Seleccionamos Alemania por la zona horaria	93
Figura 72. Dejamos la predeterminada	93
Figura 73. Si tienes un Proxy HTTP colócalo, pero si no déjalo en blanco y dale continuar.....	94
Figura 74. Esto depende de cada quien en mi caso le daré que no a la encuesta.....	94
Figura 75. Lo configuraremos de esta manera para que Debian corra en modo servidor y el ssh para usar PuTTY y controlar el servidor de mejor manera.....	95
Figura 76. Le damos en si para instalar el GRUB de arranque.....	95
Figura 77. Seleccionamos el /dev/sda para que se instale el GRUB	96
Figura 78. Le damos continuar para que se reinicie y se muestre el GRUB	96

Figura 79. Este es el GRUB de arranque de Debian 12 server	97
Figura 80. Solo iniciamos sesión y buscamos nuestra ip con #ip a y así conectarnos por PuTTY mediante ssh.....	97
Figura 81. En la aplicación PuTTY colocamos la ip	98
Figura 82. Le damos en aceptar	98
Figura 83. Aquí ya estamos dentro del servidor remotamente	99
Figura 84. Descarga de QEMU.....	99
Figura 85. Se instala Python, SPICE entre otros paquetes necesarios para la virtualización. # apt install build-essential python3 python3-pip xz-utils pkg-config libglib2.0-dev libmount-dev libpixman-1-dev libspice-protocol-dev libspice-server-dev libgail-3-dev liblz4-dev -y	100
Figura 86. Descomprimimos el archivo con # tar Jvxf qemu-8.1.0.tar.xz.....	100
Figura 87. Instalamos el paquete ninja para construir el programa de QEMU con # apt-get install ninja-build	101
Figura 88. Nos movemos a la carpeta descomprimida con el siguiente comando # cd qemu-8.1.0	101
Figura 89. primero configuramos el compilador para la construccion del software con el siguiente comando # ./configure—prefix=/opt/qemu-8 --target-list=i386-softmmu,x86_64-softmmu—enable-spice—enable-gtk.....	102
Figura 90. Primero usamos # nproc para ver cuántos núcleos tienes disponibles para ahora si usar # make -j 1 para construir la aplicación.	102

Figura 91. primero damos un # cd .. para crear el script bridge # nano bridge.sh y dentro de ese scrip llevara el comando que se ve en la figura mostrada, hay que cambiar la ip a la de ustedes, guardamos el script con ctrl + o	103
Figura 92. le aplicamos el comando de ejecución con # chmod +x bridge.sh.....	103
Figura 93. vamos a ejecutar el script # ./bridge.sh y después # ip a para ver si se realizaron los cambios.	104
Figura 94. nos movemos a la carpeta raiz # cd / despues creamos un directorio con # mkdir vm nos movemos a la carpeta creada con # cd /vm y volvemos a crear 2 directorios más como se ven en la figura.	104
Figura 95. buscamos la ruta para meterla en la variable de entorno y se pueda ejecutar, usamos el siguiente comando # find / -name qemu-img 2>/dev/null	105
Figura 96. ejecutamos # nano ~/.bashrc para abrir el PATH y asignar nuestra variable de entorno export PATH=\$PATH:/root/qemu-8.1.0/build/qemu-bundle/opt/qemu-8/bin/ guardamos con ctrl + o y cerramos con ctrl + x	105
Figura 97. Guardamos los cambios con # source ~/.bashrc	105
Figura 98. creamos un disco virtual para almacenar el sistema operativo, pero primero nos movemos a la carpeta # cd disco/ despues ejecutamos el comando # qemu-img create -f qcow2 disco1.qcow2 20G	105
Figura 99. Movemos el ISO mediante FileZilla	106
Figura 100. Movemos la ISO de home al directorio creado para poder instalar el sistema operativo en el disco virtual.....	106
Figura 101. Ejecutamos # nano equipo1.sh dentro de nuestro script ponemos.....	107



Figura 102. Solo damos permiso de ejecución con # chmod +x equipo1.sh	107
Figura 103. Ejecutamos el script.....	107
Figura 104. Ponemos la ip de la siguiente manera spice://192.168.127.133:5930.....	108
Figura 105. después ya podremos visualizar la virtualización del sistema operativo de forma remota.	108
Figura 106. creamos una copia de seguridad de la base de datos del servidor Ubuntu usamos el siguiente comando # mysqldump -u nextcloud -p nextcloud > respaldo.sql. después te pedirá la contraseña de la base de datos.	109
Figura 107. Verificamos que se haya hecho correctamente el respaldo.	109
Figura 108. Primero empaquetamos el directorio de nextcloud con el siguiente comando # tar -cvf nextcloud.tar /var/www/html/nextcloud.	109
Figura 109. Ahora comprimimos el archivo empaquetado con un nivel 9 de compresión	109
Figura 110. movemos los archivos a /home/itao con el comando # mv nextcloud.tar.xz respaldo.sql /home/itao/	110
Figura 111. Paso de archivos del servidor Ubuntu a una computadora con el respaldo de nextcloud y la base de datos.	110
Figura 112. instalacion de paquetes de php para el funcionamiento de nextcloud con los siguientes comandos # sudo apt install -y php-cli php-fpm php-json php-intl php-imagick php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml php-pear php-bcmath php-gm.....	111
Figura 113. Ahora instalamos MariaDB usando el siguiente comando # apt install mariadb-server -y	111

Figura 114. ingresamos el siguiente comando para darle seguridad a mysql # mysql_secure_installation despues le damos a todas yes e ingresamos una contraseña para volver a darle todo yes.	112
Figura 115. ahora intentamos ingresar a la base de datos con el siguiente comando y usando la contraseña que se ingresó anteriormente # mysql -u root -p	112
Figura 116. comenzamos a meter los comandos de la base de datos creamos el usuario con contraseña # CREATE USER 'itaonex'@'localhost' IDENTIFIED BY 'N3XCKVMitao123* ; creamos la base de datos # CREATE DATABASE nextcloud ; verificamos si se creó la base de datos # SHOW DATABASES ;.....	113
Figura 117. asignamos permisos al usuario # GRANT ALL PRIVILEGES ON nextcloud.* TO 'itaonex'@'localhost' ; aplicamos cambios a la base de datos # FLUSH PRIVILEGES ;.....	113
Figura 118. Pasamos los archivos respaldados a Debian.....	114
Figura 119. Importamos el respaldo de la base de datos con el siguiente comando # mysql -u itaonex -p nextcloud < /home/itaovmkvm/respaldo.sql después nos pedirá la contraseña que creamos del nuevo usuario.....	114
Figura 120. Verificamos que esten todas las tablas, primero iniciamos en la base de datos con el siguiente comando # mysql -u itaonex -p ahora nos pedira la contraseña dle usuario creado, despues ponemos # show databases ; para ver las bases de datos y escogemos nextcloud usando # use nextcloud una ves dentro de la base de datos procedemos a usar # show tables ; para ver las tablas que se importaron.	115
Figura 121. Ahora instalamos apache 2 con el siguiente comando # apt install -y apache2 libapache2-mod-php	115

Figura 122. Una vez instalado nos movemos al siguiente directorio # cd /etc/apache2/sites-available/	116
Figura 123. Creamos una copia del archivo para modificar con el siguiente comando # cp 000-default.conf nextcloud.conf	116
Figura 124. Usamos # nano ~/.bashrc para editar el path y escribimos lo siguiente # export PATH=\$PATH:/usr/sbin y guardamos el archivo y después escribimos # source ~/.bashrc para guardar los cambios.	116
Figura 125. Desactivamos 000-default.conf con # a2dissite 000-default.conf después solo reiniciamos apache 2 con # systemctl reload apache2	117
Figura 126. Editamos la siguiente línea de comando # DocumentRoot /var/www/html/nextcloud/.....	117
Figura 127. Ahora habilitamos los módulos con # a2enmod rewrite dir mime env headers después reiniciamos apache 2 # systemctl restart apache2	118
Figura 128. ingresamos la siguiente línea de comando en el archivo nextcloud.conf con # nano nextcloud.conf.....	118
Figura 129. Nos movemos a la siguiente carpeta con # cd /var/www/html/ creamos un nuevo directorio # mkdir nextcloud le damos permiso 750 # chmod 750 nextcloud/ ahora asignamos un grupo # chown www-data:www-data nextcloud/.....	119
Figura 130. Descargamos el repositorio usando # wget https://download.nextcloud.com/server/installer/setup-nextcloud.php	119
Figura 131. Asignamos un grupo al repositorio descargado.....	120

Figura 132. Ahora activamos el nextcloud.conf # a2ensite nextcloud.conf procedemos a reiniciar apache 2 # systemctl restart apache2	120
Figura 133. Verificamos que el estatus de apache 2 esté funcionando correctamente # systemctl status apache2	120
Figura 134. Entramos en el navegador usando la IP del servidor y abrimos el setup-nextcloud.php	121
Figura 135. Configuramos el nextcloud para ahora si poder remplazar los archivos necesarios.	121
Figura 136. Descomprimimos el archivo nextcloud.tar.xz # tar Jvxf nextcloud.tar.xz	121
Figura 137. primero verificamos que el archivo se descomprimió usando # ls -lh después movemos el contenido descomprimido usando # mv -n /home/itaovmkvm/nextcloud /var/www/html	122
Figura 138. Nos pedirá un PHP más antiguo así que procedemos a instalarlo.....	122
Figura 139. Descargamos el repositorio de PHP que necesitamos	123
Figura 140. Actualizamos e instalamos la versión 8.0.....	123
Figura 141. Desactivamos la versión anterior # sudo a2dismod php8.2 activamos la nueva # sudo a2enmod php8.0 procedemos a reiniciar apache 2 # sudo systemctl restart apache2	124
Figura 142. ahora solo nos falta configurar la IP en el nextcloud para terminar.....	124
Figura 143 ahora solo configuramos la siguiente ip usando # nano /var/www/html/nextcloud/nextcloud/config/config.php	125
Figura 144. Ahora ya podremos tener ejecutando correctamente el nextcloud.	125
Figura 145. Configuramos nuestro nuevo código de ápice modificado	126

Figura 146. Agregamos las ip de ejecución con sus puertos para visualizar las máquinas virtuales

126

Figura 147. Agregamos botones para cada máquina que se va ejecutar..... 127

Figura 148. Se vera de la siguiente forma para poder visualizar las maquinas 127

Figura 149. Esta es una maquina Windows donde se ejecuta Word desde el navegador..... 128

Figura 150. Aquí se está ejecutando Visual Code 128

Figura 151. Por último, ejecutamos Android Studio 129

CAPITULO 1 Tecnologías implementadas en el desarrollo

1.1 Introducción

Este proyecto se centra en la implementación de cómputo en la nube para la virtualización de aplicaciones en una nube privada. Esta solución abre las puertas para que los estudiantes del Instituto Tecnológico de Álvaro Obregón ejecuten sistemas operativos Windows y Linux de manera remota, incluso si disponen de equipos de cómputo con limitaciones.

En la base de esta implementación se encuentra la plataforma de virtualización QEMU (Quick Emulator), potenciada por KVM (Kernel Virtual Machine) para optimizar el rendimiento y la gestión de las máquinas virtuales. Estas instancias virtualizadas se albergan en una nube privada esto permite que los estudiantes accedan a sus aplicaciones como Android Studio, Visual Code.

Para administrar eficazmente estos recursos en la nube, se utiliza el sistema Nextcloud, una plataforma de colaboración y almacenamiento en la nube de código abierto que ofrece una solución integral para la gestión de archivos y aplicaciones.

El objetivo principal de esta implementación es proporcionar una solución a los estudiantes del ITAO, superando las limitaciones de sus equipos de cómputo. Esto no solo se traduce en ahorros significativos en costos, este proyecto puede servir como un modelo para otros que busquen soluciones similares.

A lo largo de este proyecto, se detallarán los pasos clave para llevar a cabo esta implementación en una nube privada, desde la configuración inicial hasta la gestión de las aplicaciones virtualizadas. Se espera que esta iniciativa no solo inspire, sino también siente las bases sólidas

para aquellos que busquen optimizar el acceso a aplicaciones informáticas esenciales en un entorno moderno y altamente conectado, especialmente en el contexto educativo.

1.2 Generalidades del proyecto

Nuestro proyecto ofrece a los estudiantes de Tecnología de Información y Comunicación en el Instituto Tecnológico de Álvaro Obregón un acceso más eficiente a recursos de cómputo. Implementaremos Cloud Computing en una nube privada, partiendo de un servidor Debian que alberga Nextcloud con el protocolo SPICE. Esta estrategia técnica incluirá el uso de QEMU y KVM para garantizar la estabilidad y ejecución de máquinas virtuales que pueden contener sistemas operativos como Windows o Linux, así como herramientas fundamentales como Android Studio, Visual Code y la paquetería de Office.

1.3 ¿Qué significa “Cloud Computing”?

“La computación en la nube y las soluciones asociadas permiten acceder, a través de la Web, a recursos y productos informáticos, que incluyen herramientas para desarrolladores, aplicaciones comerciales, servicios de computación, almacenamiento de datos y soluciones de redes. Estos servicios en la nube se alojan en un centro de datos del proveedor de software y los gestiona el proveedor de servicios en la nube o estos se gestionan en entornos locales en el centro de datos del cliente” (Oracle, 2023).

1.4 Tipos de modelos de implementación de Cloud Computing

Según Oracle (2023), se menciona: “existen tres tipos de nube: pública, privada e híbrida. Cada tipo requiere un nivel distinto de gestión por parte del cliente y ofrece un nivel de seguridad diferente” (Oracle, 2023).

1.4.1 Nube publicas

“En una nube pública, toda la infraestructura informática está ubicada en las instalaciones del proveedor de servicios en la nube, que presta estos servicios al cliente a través de Internet. Los clientes no tienen que mantener sus propios recursos informáticos y pueden agregar rápidamente más usuarios o potencia informática según sea necesario. En este modelo, diversos inquilinos comparten la infraestructura informática del proveedor de la nube” (Oracle, 2023).

1.4.2 Nube privada

“La nube privada es de uso exclusivo de una organización. Se puede alojar en la ubicación de la organización o en el centro de datos del proveedor de la nube. La nube privada proporciona un nivel más alto de seguridad y control” (Oracle, 2023).

1.4.3 Nube híbrida

“Como su nombre indica, una nube híbrida es una combinación de nubes públicas y privadas. En general, los clientes de nubes híbridas alojan sus aplicaciones críticas en sus propios servidores para mayor seguridad y control, y las aplicaciones secundarias se almacenan en el entorno de un proveedor de nube” (Oracle, 2023).

1.4.4 Nube multiclouds (Red Hat, 2023)

“El término multicloud se refiere a un enfoque de nube compuesto por al menos dos servicios de nube, que proporcionan por lo menos dos proveedores de nube pública o privada. Todas las nubes híbridas son multiclouds, pero no todas las multiclouds son híbridas. Las multiclouds se vuelven híbridas cuando se conectan varias nubes con algún tipo de integración u organización”.

“Un entorno multicloud puede crearse a propósito (para controlar mejor los datos confidenciales o como un espacio de almacenamiento redundante para una mejor recuperación ante desastres)”.

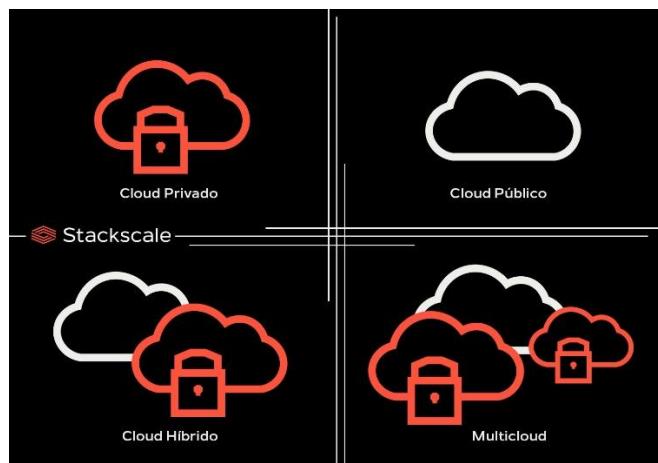


Figura 1. (Se muestra cómo se ven los diferentes modelos de cloud computing.) (Stackscale, 2023)

En la tabla 1 se destaca claramente que Amazon Web Services (AWS) se posiciona como el líder indiscutible en la adopción de todos los modelos de nube, abarcando una amplia gama de necesidades para sus diversos clientes.



Tabla 1. Empresas en la Adopción de Modelos de Implementación de Cloud Computing

Modelo de Implementación	Empresas	Puntos Destacados
Nube Pública	Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP)	Escalabilidad, bajo costo inicial, amplia gama de servicios.
Nube Privada	VMware, IBM Cloud Private, Oracle Cloud at Customer	Mayor control de datos y seguridad, personalización.
Nube Híbrida	AWS Outposts, Azure Hybrid Cloud, Google Anthos	Flexibilidad para mover cargas de trabajo, equilibrio entre nube pública y privada.
Nube Comunitaria	Google Government Cloud, Oracle Cloud for Research, AWS GovCloud (US)	Cumplimiento normativo, colaboración entre organizaciones.
Nube Multicloud	Salesforce (utiliza AWS y Azure), Dropbox (utiliza AWS y Google Cloud), Netflix (utiliza AWS y Google Cloud)	Redundancia y mitigación de riesgos, elección de la mejor plataforma para cada carga de trabajo.

1.5 Tipos de servicios de Cloud Computing

Según Red Hat, se menciona: “Los servicios de nube son infraestructuras, plataformas o sistemas de software que los proveedores externos alojan y ponen a disposición de los usuarios a través de Internet” (Red Hat, 2023).

1.5.1 Infraestructura como servicio (IaaS)

“El modelo de Infraestructura como servicio (IaaS) contiene los componentes básicos de la informática en la nube y proporciona normalmente acceso a las funciones de red, ordenadores (hardware virtual o dedicado) y espacio de almacenamiento de datos. IaaS le ofrece el mayor grado de flexibilidad y control de administración sobre sus recursos de TI y, en general, es similar a los recursos de TI existentes que muchos departamentos de TI y desarrolladores ya conocen” (Amazon Web Service, 2023).

1.5.2 Plataforma como servicio (PaaS)

“El modelo de Plataforma como servicio (PaaS) permite a su organización prescindir de la administración de la infraestructura subyacente (normalmente, hardware y sistemas operativos) y dedicarse a la implementación y administración de sus aplicaciones. Esto contribuye a mejorar su eficacia, pues no tiene que preocuparse del aprovisionamiento de recursos, la planificación de la capacidad, el mantenimiento de software, los parches ni ninguna de las demás arduas tareas que conlleva la ejecución de su aplicación” (Amazon Web Service, 2023).

1.5.3 Software como servicio (SaaS)

“El modelo de Software como servicio (SaaS) le ofrece un producto completo ejecutado y administrado por el proveedor del servicio. En la mayoría de los casos, quienes hablan de Software

como servicio en realidad se refieren a aplicaciones de usuario final. Con una solución SaaS, no tiene que preocuparse de cómo se mantendrá el servicio o cómo se administrará la infraestructura subyacente; solo necesita determinar cómo va a usar ese componente específico de software. Un ejemplo común de una aplicación SaaS es un sistema de correo electrónico por Internet que puede usar para enviar y recibir correo electrónico sin tener que administrar la incorporación de características en el producto de correo electrónico ni mantener los servidores y sistemas operativos en los que se ejecuta el programa” (Amazon Web Service, 2023).

1.5.4 Función como servicio (FaaS)

“Es un modelo de ejecución basado en eventos que permite que los desarrolladores diseñen, ejecuten y gestionen paquetes de aplicaciones como funciones, sin tener que preocuparse por el mantenimiento de la infraestructura” (Red Hat, 2023).

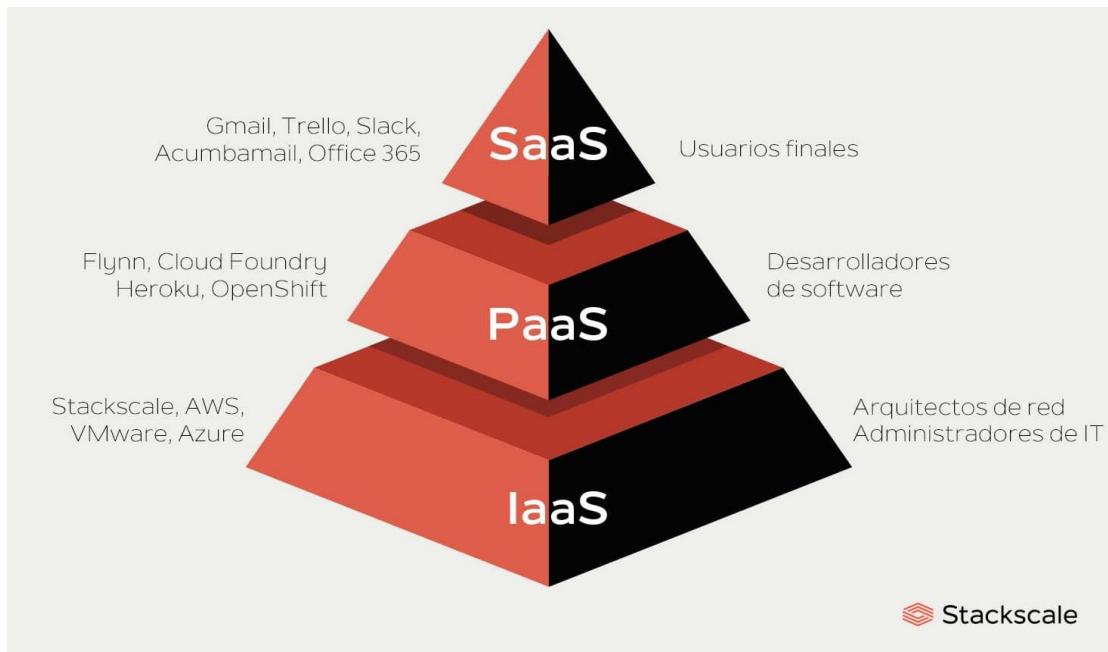


Figura 2. “Cada modelo de servicio en la nube cubre diferentes necesidades de usuarios y empresas” (Stackscale, 2023)

Tabla 2 Empresas de Tecnología y sus tipos de Nube

Empresa	Tipo de Servicio	Modelo de Implementación	Sistemas Operativos Soportados
Amazon Web Services	IaaS, PaaS, SaaS, FaaS	Pública	Windows, Linux
Microsoft Azure	IaaS, PaaS, SaaS, FaaS	Pública	Windows, Linux
Google Cloud Platform	IaaS, PaaS, SaaS, FaaS	Pública	Windows, Linux
Heroku	PaaS	Pública	Variados, con soporte para múltiples lenguajes de programación.
Google App Engine	PaaS	Pública	Variados, con soporte para múltiples lenguajes de programación.
Red Hat OpenShift	PaaS	Privada	Linux
Salesforce	SaaS	Pública	Interfaz web, compatible con múltiples sistemas operativos cliente.

Empresa	Tipo de Servicio	Modelo de Implementación	Sistemas Operativos Soportados
Microsoft 365	SaaS	Pública	Interfaz web, compatible con múltiples sistemas operativos cliente.
Dropbox	SaaS	Pública	Interfaz web, compatible con múltiples sistemas operativos cliente.
AWS Lambda	FaaS	Pública	Linux.
Azure Functions	FaaS	Pública	Windows.
Google Cloud Functions	FaaS	Pública	Linux.

En la tabla 2 se evidencia la preferencia de las empresas por ciertos servicios sobre otros, así como la distribución de sistemas operativos utilizados por estas empresas en sus servidores.

1.6 Tipos de hipervisor

Los hipervisores, también conocidos como VMM (Virtual Machine Monitor), se dividen en dos categorías principales: **Tipo 1 (Bare-Metal)** y **Tipo 2 (Hosted)**. Estos son software o firmware que permiten la virtualización de hardware y la creación de máquinas virtuales dentro de un único servidor, lo que permite ejecutar diferentes sistemas operativos de manera aislada.

1.6.1 Hipervisor de Tipo 1

“Un hipervisor de tipo 1 se ejecuta directamente en el hardware físico de la computadora subyacente, interactuando directamente con su CPU, memoria y almacenamiento físico. Por esta razón, los hipervisores de Tipo 1 también se denominan hipervisores bare-metal. Un hipervisor de Tipo 1 ocupa el lugar del sistema operativo host” (IBM, 2023).

1.6.1.1 Hipervisores nativos

Estos hipervisores se ejecutan directamente en el hardware físico del servidor y no requieren un sistema operativo host. Ejemplos incluyen VMware vSphere/ESXi y Microsoft Hyper-V.

1.6.1.2 Hipervisores de partición

Estos hipervisores dividen el sistema físico en particiones virtuales y permiten que cada partición se comporte como una máquina física independiente. Ejemplos incluyen Xen y Oracle VM Server for SPARC/x86.

1.6.1.3 Hipervisor Hibrido

Este hipervisor híbrido utiliza QEMU, que es de Tipo 2, en conjunto con KVM, que pertenece a la categoría Tipo 1. Al combinar estas tecnologías, logra comportarse como un hipervisor de Tipo 1 en términos prácticos, lo que se traduce en un rendimiento casi nativo y un acceso directo al hardware del servidor. Estas características ofrecen un alto rendimiento y una sólida seguridad, típicas de los hipervisores de Tipo 1.

“Linux también tiene capacidades de hipervisor integradas directamente en el kernel de su sistema operativo. La máquina virtual basada en kernel (KVM) se convirtió en parte de la línea

principal del kernel de Linux en 2007 y complementa QEMU, que es un hipervisor que emula el procesador de la máquina física completamente en software” (IBM, 2023).

1.6.2 Hipervisor de Tipo 2

“Un hipervisor de Tipo 2 no se ejecuta directamente en el hardware subyacente. En cambio, se ejecuta como una aplicación en un sistema operativo. Los hipervisores de Tipo 2 rara vez aparecen en entornos basados en servidor. En cambio, son adecuados para usuarios de PC individuales que necesitan ejecutar varios sistemas operativos. Los ejemplos incluyen ingenieros, profesionales de la seguridad que analizan malware y usuarios comerciales que necesitan acceso a aplicaciones que solo están disponibles en otras plataformas de software” (IBM, 2023).

1.6.2.1 Hipervisores de emulación

Estos hipervisores emulan completamente el hardware de la computadora, lo que los hace versátiles, pero menos eficientes en términos de rendimiento. Ejemplos incluyen QEMU y Bochs.

1.6.2.2 Hipervisores de virtualización ligera

Estos hipervisores permiten la ejecución de máquinas virtuales en un sistema operativo host, compartiendo recursos con el sistema operativo anfitrión para lograr un mejor rendimiento que la emulación completa. Ejemplos incluyen VMware Workstation y VirtualBox.

1.6.3 Otros Subtipos de Hipervisores

1.6.3.1 Hipervisores para contenedores

Estos hipervisores se centran en la virtualización a nivel de sistema operativo y permiten la creación de contenedores, como Docker. Ejemplos incluyen Docker y Kubernetes.



1.6.3.2 Hipervisores de administración de clústeres

Estos hipervisores están diseñados para administrar clústeres de máquinas virtuales en entornos de centro de datos. Ejemplos incluyen VMware vCenter Server y Microsoft System Center Virtual Machine Manager.

1.6.3.3 Hipervisores de código abierto

Estos hipervisores son software de virtualización de código abierto y pueden estar disponibles en varias categorías mencionadas anteriormente. Ejemplos incluyen KVM, Xen y VirtualBox.

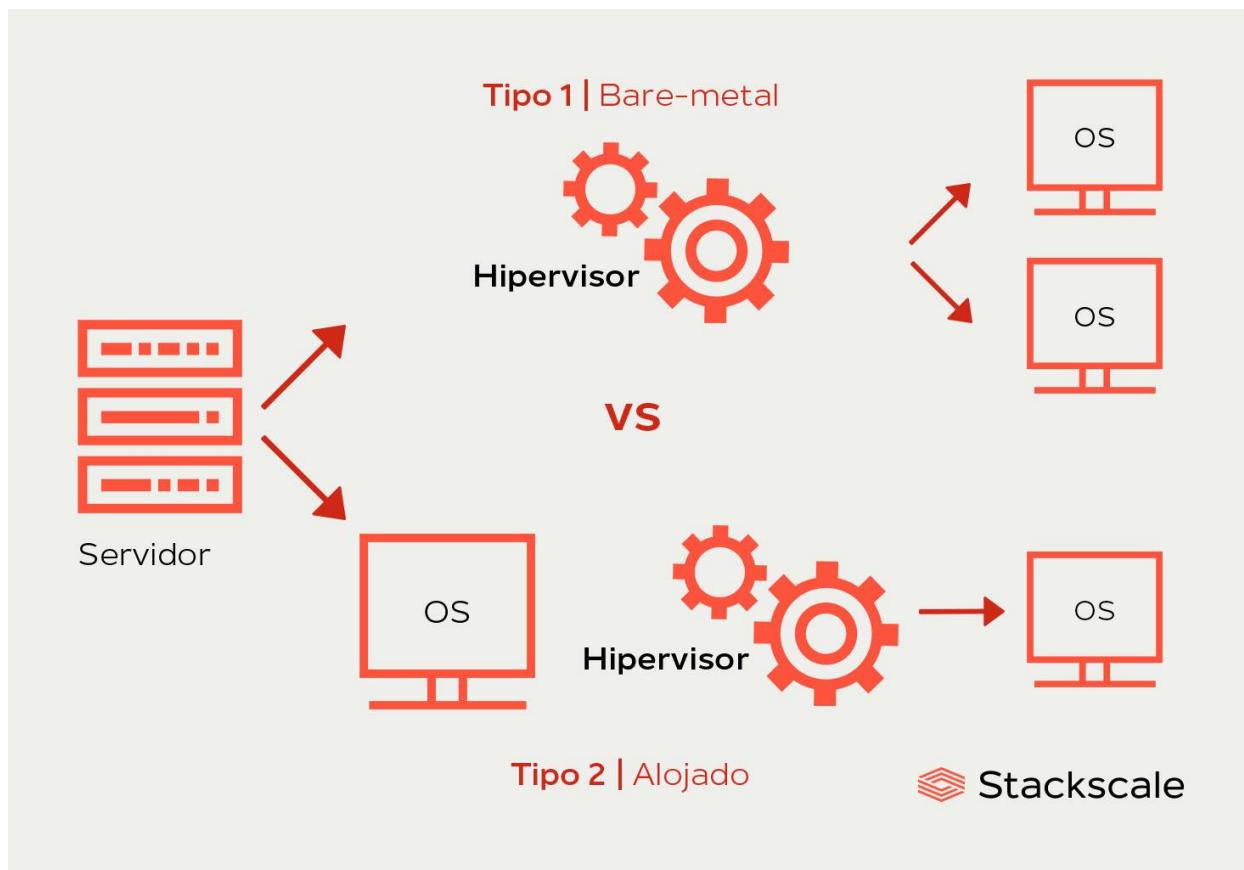


Figura 3. Comparación de hipervisores de tipo 1 y tipo 2 (Stackscale, 2023).

Tabla 3. Comparación entre Hipervisores Tipo 1 y Tipo 2

Aspecto	Hipervisor Tipo 1 (Bare-Metal)	Hipervisor Tipo 2 (Hosted)
Instalación	Se instala directamente en el hardware físico del servidor.	Se instala como una aplicación dentro de un sistema operativo host existente.
Dependencia del SO Host	No depende de un sistema operativo host.	Dependiente de un sistema operativo host que gestiona recursos.
Rendimiento	Ofrece un rendimiento cercano al nativo al acceder directamente al hardware.	Puede tener un rendimiento ligeramente inferior debido a la intermediación del sistema operativo host.
Recursos compartidos	Recursos compartidos mínimos con otros sistemas en el mismo hardware.	Comparte recursos con el sistema operativo host y otras aplicaciones en ejecución.
Seguridad	Mayor seguridad, ya que se encuentra en una capa más profunda y aislada.	La seguridad depende del sistema operativo host y puede ser menos sólida.
Ejemplos	VMware vSphere/ESXi, Microsoft Hyper-V, Xen	VMware Workstation, VirtualBox, Parallels Desktop

En la tabla 3, se aprecia la distinción entre los hipervisores de tipo 1 y tipo 2, así como sus respectivos modos de funcionamiento y características específicas.



1.7 PLANTEAMIENTO DEL PROBLEMA

La pandemia de COVID-19 en 2019 resaltó la carencia de recursos informáticos adecuados para los estudiantes del Instituto Tecnológico de Álvaro Obregón. Inicialmente, implementamos Nextcloud en una nube privada en un servidor local del ITAO. Aunque esta solución brindó apoyo durante la pandemia, los desafíos relacionados con los recursos persistieron.

Para superar esta limitación, fusionamos Nextcloud con la virtualización de sistemas Windows y Linux, lo que permitirá a los estudiantes acceder a herramientas como Android Studio y Visual Studio Code mediante un servicio SaaS en un entorno local del ITAO.

1.8 OBJETIVO GENERAL

Proporcionar una solución de cómputo en la nube de tipo ‘nube privada’, destinada a la virtualización de aplicaciones, con el propósito de usar programas de la carrera TICs como Android Studio, Visual code. Esta solución se enfoca en mejorar las aplicaciones, brindando a los estudiantes la capacidad de acceder y utilizar recursos informáticos de manera óptima, incluso en situaciones donde los recursos de hardware son limitados.

1.9 OBJETIVO ESPECIFICO

Implementar cómputo en la nube para la virtualización de sistemas operativos Windows o Linux.

Nextcloud para alojamiento de virtualización de aplicaciones en la nube mediante QEMU y KVM.

Ejecutar aplicaciones en equipos de cómputo no dedicados mediante cómputo en la nube.

1.10 JUSTIFICACION

El ITAO enfrenta un desafío en cuanto a la disponibilidad de equipos potentes para sus alumnos.

La solución propuesta permitirá a los alumnos del ITAO acceder a aplicaciones desde cualquier dispositivo de manera local, eliminando limitaciones de hardware. Esto habilitará la ejecución de aplicaciones en equipos no dedicados.

La virtualización de aplicaciones permitirá el acceso a software especializado desde cualquier lugar y momento de manera local, mejorando la eficiencia académica y profesional de los alumnos.

La seguridad de datos será una prioridad, con medidas sólidas implementadas para proteger la información personal de los alumnos y garantizar el acceso solo a usuarios autorizados.

1.11 DELIMITACIONES

Alcance de la Virtualización: El proyecto se enfocará únicamente en la virtualización de aplicaciones Windows mediante cómputo en la nube.

Tiempo Estimado: El proyecto tendrá una duración estimada de 4 a 6 meses. Se debe cumplir con los objetivos establecidos dentro de este marco temporal.

Espacio y Territorio: El proyecto se llevará a cabo en la Ciudad de México, específicamente en la alcaldía Álvaro Obregón, donde se encuentra el Instituto Tecnológico de Álvaro Obregón (ITAO).

Destinatarios: El enfoque del proyecto estará dirigido a los alumnos del ITAO, quienes serán los principales beneficiarios de la solución de cómputo en la nube.

Delimitación de Recursos: La infraestructura de cómputo en la nube se basará en un servidor Dell FTYWJ93 POWEREDGE R440, con un procesador Xeon Silver 4110 y 8 GB de RAM, así como un almacenamiento de SSD de 500GB Kingston.

CAPITULO 2 MARCO CONCEPTUAL

2.1 DEBIAN 12 SERVER (Debian, 2023)

Debian 12, también conocido como “Bookworm”, es la última versión estable del sistema operativo Debian GNU/Linux. Esta versión es relevante en el contexto de tu proyecto de implementación de cómputo en la nube, ya que Debian es una distribución de Linux ampliamente respetada y utilizada en entornos de servidores y estaciones de trabajo.

Soporte para múltiples arquitecturas: Debian 12 es compatible con varias arquitecturas, lo que significa que puede ejecutarse en una variedad de hardware, incluyendo sistemas de 64 bits, sistemas ARM de 64 bits, sistemas de 32 bits y más. Esto brinda flexibilidad en la elección de hardware para tu infraestructura de cómputo en la nube.

Amplia colección de software: Debian es conocido por su vasta colección de software disponible. Esto es beneficioso para tu proyecto, ya que puedes acceder a una amplia gama de aplicaciones y herramientas que pueden ser útiles en tu entorno de nube.

Mejoras en el soporte de hardware y actualizaciones de paquetes: El soporte mejorado de hardware es esencial para garantizar que tu infraestructura de cómputo en la nube funcione sin problemas en una variedad de sistemas. Además, las actualizaciones de paquetes mantendrán tu sistema seguro y actualizado.

Enfoque en la seguridad y la estabilidad: Debian es conocido por su enfoque en la seguridad y la estabilidad del sistema. Esto es crucial cuando estás implementando una solución de cómputo

en la nube, ya que garantiza que tus datos y aplicaciones estén protegidos y que el sistema funcione de manera confiable.

Tabla 4 Características Destacadas de Debian 12

Característica	Descripción en Debian 12
Arquitecturas	64-bit PC (amd64), 64-bit ARM (AArch64), 32-bit PC (i386)
Entornos de escritorio	GNOME, KDE Plasma, Xfce, LXDE, LXQt, MATE, Cinnamon
Gestor de paquetes	APT (Advanced Package Tool)
Sistema de inicio	systemd 252
Seguridad	AppArmor, SELinux, seccomp
Virtualización	KVM, QEMU, Xen, LXC, Docker
Contenedores	Docker, Podman, LXC/LXD
Servidores web	Apache, Nginx, Lighttpd
Bases de datos	MariaDB, PostgreSQL, SQLite, Redis
Lenguajes de programación	Python, Perl, Ruby, PHP, Java, Rust, Go, Node.js

En la tabla 4 se detallan las características de Debian en relación con distintos tipos de servicios,

2.2 QEMU (QEMU, 2023)

QEMU, abreviatura de Quick Emulator, es una herramienta de código abierto que combina emulación y virtualización para ejecutar sistemas operativos y aplicaciones en una amplia variedad de arquitecturas. Desarrollado por Fabrice Bellard, QEMU es compatible con sistemas operativos como Linux, Windows y macOS.

QEMU opera en dos modos distintos: emulación de hardware y virtualización. En el modo de emulación, simula todos los componentes de hardware, permitiendo la ejecución de sistemas y programas diseñados para arquitecturas diferentes a la del sistema anfitrión. Por ejemplo, es posible ejecutar software ARM en una PC x86.

En el modo de virtualización, QEMU permite ejecutar múltiples máquinas virtuales en una única máquina física, lo que maximiza la eficiencia de los recursos y reduce costos. Esto es especialmente útil en entornos de servidores.

QEMU admite una amplia gama de arquitecturas, incluyendo x86, x86-64, PowerPC, MIPS y SPARC. Aunque no proporciona una interfaz gráfica de usuario (GUI) por defecto, existen programas adicionales como QEMU Manager y qemu-launcher que ofrecen GUI para simplificar la gestión de máquinas virtuales.

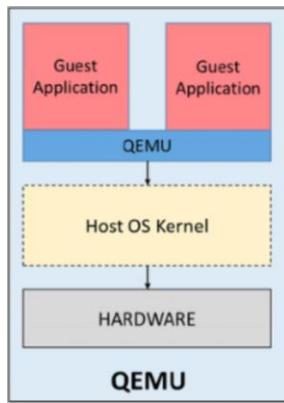


Figura 4. Arquitectura del funcionamiento de QEMU.

Tabla 5. Características de QEMU

Característica	Descripción en QEMU
Emulador y virtualizador	QEMU permite emular procesadores de computadoras mediante la traducción dinámica de binarios en sistemas operativos Unix y Linux.
Administración remota	QEMU se puede administrar de forma remota utilizando herramientas como PuTTY para establecer conexiones SSH seguras a servidores.
Soporte para múltiples arquitecturas	QEMU es compatible con una amplia variedad de arquitecturas, incluyendo x86, x86-64, PowerPC, MIPS y SPARC.

En la tabla 5 se presentan las características de QEMU y sus diversas utilidades dentro de este proyecto.

2.3 KVM (Red Hat, 2023)

“Las máquinas virtuales basadas en el kernel (KVM) son una tecnología de virtualización open source integrada a Linux®. Con ellas, puede transformar Linux en un hipervisor que permite que una máquina host ejecute varios entornos virtuales aislados llamados máquinas virtuales (VM) o guests”.

“Las KVM forman parte de Linux. Por eso, si cuenta con una versión de Linux 2.6.20 o posterior, ya las tiene a su disposición. Se anunciaron por primera vez en 2006, y un año después se incorporaron a la versión principal del kernel de Linux. Dado que forman parte del código actual de Linux, reciben inmediatamente todas las mejoras, las correcciones y las funciones nuevas de este sistema, sin requerir ningún tipo de ingeniería adicional”.

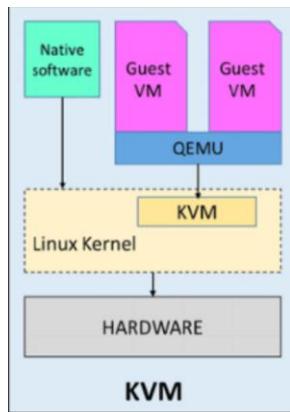


Figura 5. Arquitectura del funcionamiento de KVM.

Tabla 6. Características de Kernel-based Virtual Machine

Característica	Descripción en KVM
Tipo de virtualización	KVM es una solución de virtualización de código abierto que se integra directamente en el kernel de Linux, proporcionando virtualización completa y para virtualización.
Compatibilidad con sistemas operativos	KVM es compatible con múltiples sistemas operativos, incluyendo Linux, Windows y otros.
Rendimiento	KVM ofrece un alto rendimiento gracias a su integración en el kernel de Linux y su capacidad para realizar virtualización cercana al hardware del servidor.
Seguridad	KVM hereda las características de seguridad del kernel de Linux y puede implementar medidas de seguridad adicionales, como SSH, SSL, RADIUS y un firewall configurable.
Gratis y de código abierto	KVM es gratuito y de código abierto, lo que significa que las empresas no tienen que pagar licencias adicionales para alojar máquinas virtuales.
Administración remota	KVM permite la administración remota de servidores y dispositivos de red a través de conexiones IP.

Característica	Descripción en KVM
Soporte para hardware	KVM es compatible con una amplia variedad de hardware y utiliza QEMU para emular hardware y mejorar la velocidad mediante para virtualización.

Tabla 6 se muestran las características y funciones de KVM implementado en Linux.

2.4 APACHE2 (Apache, 2023)

Apache2 es un servidor web de código abierto ampliamente utilizado y respaldado por la Apache Software Foundation. Desde su lanzamiento en 1996, Apache ha desempeñado un papel crucial en el crecimiento y desarrollo de la World Wide Web.

Entre las características destacadas de Apache se encuentra su arquitectura modular, que permite la activación y desactivación de funcionalidades según las necesidades del usuario. Algunos ejemplos de módulos incluyen mod_security para la seguridad, Varnish para la caché y mod_headers para la personalización de cabeceras. Además, Apache brinda la flexibilidad de ajustar parámetros de PHP de manera personalizada mediante el archivo .htaccess.

Las ventajas de Apache abarcan su gratuidad, código abierto, facilidad de instalación y configuración, extensibilidad mediante módulos, así como funciones incorporadas para autenticación y validación de usuarios. También ofrece soporte para varios lenguajes, como Perl, PHP y Python.

2.5 NEXTCLOUD (Nextcloud, 2023)

Nextcloud es una plataforma de software de código abierto diseñada para crear y utilizar servicios de almacenamiento de archivos y colaboración en línea. Similar a servicios comerciales como Dropbox, Office 365 o Google Drive, Nextcloud brinda a los usuarios la capacidad de alojar sus datos en la nube o en servidores locales, garantizando el control total sobre la información.

- Auto alojamiento: Los usuarios tienen control total sobre sus datos, lo que les permite aplicar sus propias políticas de seguridad.
- Seguridad: Nextcloud se centra en la protección de datos del usuario, con múltiples capas de seguridad.
- Compatibilidad con bases de datos: Es compatible con bases de datos SQL estándar de la industria como PostgreSQL, MySQL y MariaDB para el almacenamiento de usuarios y metadatos.
- Colaboración en tiempo real: Facilita la colaboración en documentos, chat y videollamadas.
- Acceso multiplataforma: Ofrece aplicaciones para Android, iOS y sistemas de escritorio como Windows, Mac y Linux.

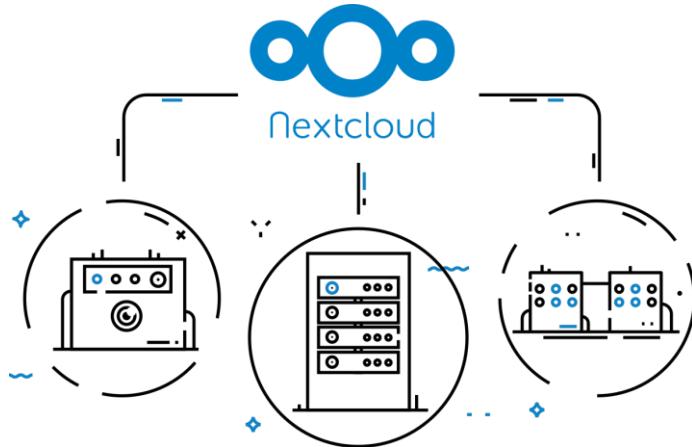


Figura 6. Arquitectura de Nextcloud

2.6 Script .sh en Bash (Hostiger, 2023)

Son archivos de texto que contienen una secuencia de comandos y operaciones ejecutados por el intérprete Bash. Estos scripts son ideales para automatizar tareas repetitivas o complejas mediante la combinación de comandos y estructuras de control en un único archivo. Por lo general, tienen la extensión .sh y comienzan con un shebang (#!), que especifica qué shell debe interpretar el script.

Para crear un script Bash, es necesario comprender los comandos del shell y su sintaxis adecuada. Puedes utilizar un editor de texto para escribir los comandos y, posteriormente, guardar el archivo con la extensión .sh, como “sampleFunction.sh” en un directorio de tu elección.

Una vez creado, puedes ejecutar el script desde la terminal mediante el comando **bash** seguido de la ruta del archivo. Por ejemplo, para ejecutar un script llamado “function.sh” en el directorio “Test/Bash”, utilizarías el comando **bash Test/Bash/function.sh**.

Los scripts Bash son valiosos para automatizar tareas de desarrollo, como la compilación de código, la depuración y la gestión de cambios. También simplifican tu flujo de trabajo al combinar comandos largos en un solo archivo ejecutable. Además, dado que Bash es el intérprete predeterminado en muchos sistemas GNU/Linux y sistemas Unix, estos scripts son altamente compatibles.

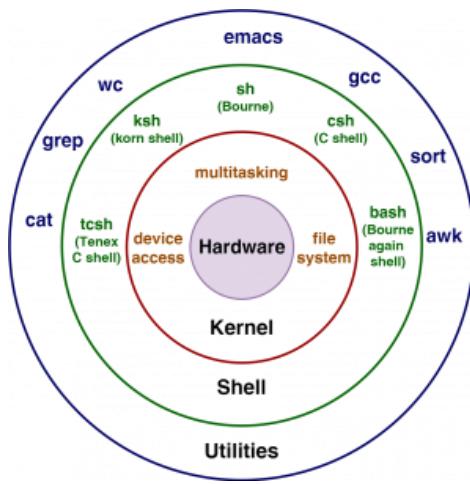


Figura 7. Arquitectura de terminal con el kernel

2.7 BRIDGE (CISCO, 2023)

Bridges en IP (Puentes de Red) Son dispositivos que operan en la capa 2 del modelo OSI (nivel de enlace de datos) y se utilizan para dividir o conectar segmentos de red dentro de una misma red lógica, como un único subconjunto de IP. Su función principal es analizar la dirección de destino de los paquetes de datos y decidir si deben ser transmitidos a otro segmento de la red. Esta operación mejora la velocidad y reduce las colisiones en la red.

Los bridges no toman decisiones basadas en el tipo de protocolo (IP, IPX, NetBEUI), sino en las direcciones MAC únicas de las tarjetas de red (NIC). Además, pueden superar

incompatibilidades de hardware sin salir del rango de direcciones de la red IP o subred, lo que permite la conexión de diferentes medios físicos, como 10 Base T y 100 Base TX.

En el caso de los puentes virtuales, actúan como dispositivos virtuales de red que dirigen paquetes entre dos o más segmentos de red. Un puente virtual se comporta como un conmutador de red virtual y opera de manera transparente. Para crear un puente virtual y activarlo, se utilizan comandos como “ip link add name bridge_name type bridge” y “ip link set dev bridge_name up”.

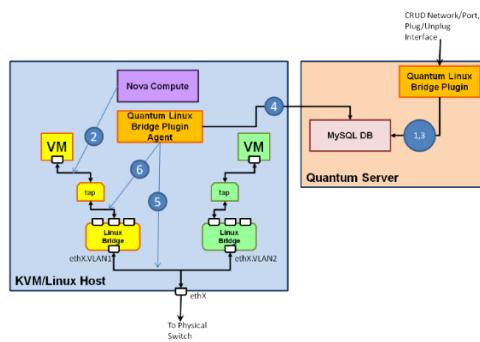


Figura 8. Arquitectura del funcionamiento de Bridge con KVM

2.8 .ISO

Archivos ISO Son imágenes de discos ópticos (CD, DVD o Blu-ray) que contienen copias exactas de sistemas de archivos, programas y datos. Los archivos ISO son ampliamente utilizados para almacenar y distribuir sistemas operativos, software, películas, videos y juegos de computadora (INOS, 2023).

2.9 SERVIDOR Dell PowerEdge R440 (DELL, 2023)

Procesador: Este servidor en rack está equipado con un procesador Intel Xeon Silver 4110. El procesador cuenta con 8 núcleos y 16 hilos, una frecuencia base de 2.1 GHz y una caché de 11 MB, ofreciendo un excelente rendimiento y eficiencia energética para aplicaciones de servidor.

Memoria RAM: El servidor incluye 8 GB de RAM, con la posibilidad de escalar hasta 1 TB.

Esto lo hace adecuado para una amplia variedad de cargas de trabajo, desde aplicaciones de alto rendimiento hasta despliegues web.

Escalabilidad: El PowerEdge R440 es altamente escalable en términos de memoria, almacenamiento y opciones de E/S. Puede admitir hasta diez unidades y ofrece flexibilidad en cuanto a configuraciones de memoria, lo que permite adaptarse a las necesidades de la infraestructura en crecimiento.

Aplicaciones: Diseñado para aplicaciones de alto rendimiento y despliegues web, el servidor PowerEdge R440 es una solución versátil que puede satisfacer una variedad de necesidades empresariales.

Compatibilidad: Es compatible con múltiples arquitecturas, incluyendo x86, x86-64, PowerPC, MIPS y SPARC, y ofrece soporte para una amplia selección de software y entornos de escritorio.

2.10 PuTTY (Wordpress, 2023)

Es un cliente de código abierto ampliamente utilizado para protocolos de acceso remoto como SSH, Telnet, rlogin y TCP raw. Desarrollado principalmente para Windows, también se ha adaptado a sistemas Unix-like y se encuentra en desarrollo para Mac OS.

- Almacenamiento de Hosts y Preferencias: PuTTY permite guardar configuraciones de hosts y preferencias para un uso futuro más sencillo.

- Control Avanzado: Ofrece control sobre la clave de cifrado SSH, la versión del protocolo y el redireccionamiento de puertos, incluyendo reenvío X11.
- Clientes de Línea de Comandos: Incluye clientes SCP y SFTP llamados “pscp” y “psftp” para transferencia segura de archivos.
- Emuladores de Terminal: Proporciona emuladores de terminal completos como xterm, VT102 y ECMA-48 para una experiencia de terminal eficaz.
- Soporte IPv6: Es compatible con el protocolo IPv6, lo que garantiza la conectividad en redes modernas.
- Soporte de Criptografía: Ofrece soporte para una amplia gama de algoritmos de cifrado, incluyendo 3DES, AES, RC4, Blowfish y DES.

2.11 FileZilla (FileZilla, 2023)

Es un cliente FTP de código abierto y gratuito que permite a los usuarios transferir archivos de manera eficiente entre computadoras a través de Internet. Es compatible con Windows, Mac OS y Linux.

- Soporte FTP y FTPS: FileZilla es compatible tanto con FTP estándar como con FTP sobre SSL/TLS (FTPS) para conexiones seguras.
- Soporte IPv6: Garantiza conectividad en redes modernas con soporte para IPv6.
- Configuración de Velocidad: Ofrece límites de velocidad configurables para adaptarse a las necesidades de transferencia.
- Manejo de Archivos Grandes: Puede manejar archivos de más de 4 GB sin problemas.

- Administración Remota: Proporciona funciones de administración remota para un control completo.
- Sistema de Permisos: Cuenta con un sistema de permisos que incluye usuarios y grupos para una gestión de acceso precisa.
- Filtros de IP: Permite configurar filtros de IP para controlar el acceso.

2.12 Protocolo SPICE

SPICE (Simple Protocol for Independent Computing Environments) es un protocolo de comunicación para entornos virtuales que permite a los usuarios acceder a la consola de máquinas virtuales (VM) desde cualquier lugar a través de Internet. Fue desarrollado por Qumranet, que luego fue adquirida por Red Hat. SPICE se utiliza principalmente en soluciones de virtualización de escritorio y se centra en proporcionar una experiencia de usuario de alta calidad al acceder a escritorios virtuales.

Tabla 7. Características del protocolo SPICE

Característica	Descripción en SPICE
Protocolo	SPICE es un protocolo de comunicación para entornos virtuales
Arquitectura	Uso de cliente (como remote-viewer) y servidor SPICE para compartir VM.
Funcionalidades	Reproducción de video, grabación de audio y compartición de dispositivos y carpetas.

Característica	Descripción en SPICE
Compatibilidad	Compatible con QEMU, KVM, Red Hat Virtualization y oVirt.
Seguridad	Utiliza cifrado TLS para proteger las comunicaciones entre cliente y servidor.

Tabla 7 se muestra el uso del protocolo SPICE y eficiencia ante virtualización

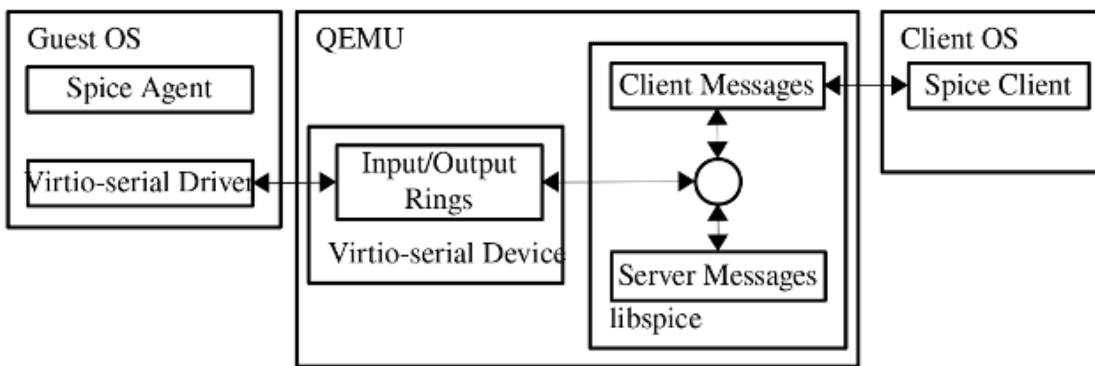


Figura 9. Arquitectura del funcionamiento del protocolo SPICE



CAPITULO 3 METOLOGÍA DevOps

3.1 Metodología DevOps

Para implementar Cloud Computing en el Instituto Tecnológico de Álvaro Obregón, se ha optado el enfoque DevOps, un ciclo de vida de desarrollo y despliegue de software creado por Patrick Debois. Esta metodología ha sido exitosamente implementada por empresas líderes como Amazon Web Service y Red Hat, entre otras. El proceso de DevOps abarca diversas fases cruciales que incluyen Desarrollo, Prueba, Integración, Despliegue y Monitoreo, asegurando así una gestión eficiente y continua de todo el entorno de Cloud Computing.



Figura 10. Ciclo DevOps (datascientest, 2023)

3.2 Desarrollo

En esta etapa, se procederá con la identificación de los requisitos mínimos de Debian y QEMU como se muestra en las tabla 8 y 9 a continuación.

Tabla 8. Requerimientos mínimos de instalación de Debian 12 Server

Sistema Operativo	Requisitos Mínimos
Debian	<ul style="list-style-type: none"> - Procesador: 1 GHz o superior - Memoria RAM: 512 MB - Almacenamiento: 10 GB de espacio en disco - Conexión a Internet - Unidad de DVD-ROM (solo para instalación desde DVD) - Interfaz gráfica de usuario o modo texto - Teclado y mouse compatibles con el sistema

Tabla 9. Requerimientos Mínimos de instalación del Hipervisor QEMU

Hipervisor QEMU	Requisitos Mínimos
Linux	Procesador de 1 GHz o superior, 1 GB de RAM o más, Al menos 5 GB de espacio en disco duro
SPICE	Procesador de 1,5 GHz o superior, 2 GB de RAM o más, Al menos 10 GB de espacio en disco duro

Hipervisor QEMU	Requisitos Mínimos
KVM	Procesador de doble núcleo o superior, 2 GB de RAM o más, Al menos 20 GB de espacio en disco duro
Servidor web	Procesador de 2 GHz o superior, 4 GB de RAM o más, Al menos 50 GB de espacio en disco duro

Dado los requerimientos mínimos de instalaciones las tablas anteriores, podemos determinar que contamos con los recursos suficientes para implementar la solución del hipervisor QEMU. Se utilizó una computadora de escritorio que, hacia la función de un servidor para las primeras pruebas, con los siguientes características y recursos de hardware y software se muestra en la tabla 10 el hardware ocupado para pruebas y errores.

Tabla 10. Requerimientos de Hardware y Software de la prueba y error

Hardware	Software
Lenovo Desktop modo servidor Procesador: AMD A10 Memoria RAM: 8GB Disco Duro: 500GB	Sistema Operativo: Debian Server 12.1 Software SaaS: QEMU, KVM y SPICE Servidor WEB: Apache2 Base de Datos: María DB

Hardware	Software
	PHP: 8.0

Posteriormente, se migro el servidor de prueba a un servidor real, donde se determinan las características en la siguiente tabla 11.

Tabla 11. Requerimientos de Hardware y software del servidor principal

Hardware	Software
<p>Servidor: DELL EMC R440 Power Edge</p> <p>Procesador: Intel Xeon de 8 núcleos de 64bits</p> <p>Memoria RAM: 8GB</p> <p>Disco Duro: 500GB</p>	<p>Sistema Operativo: Debian Server 12.1</p> <p>Software SaaS: QEMU, KVM y SPICE</p> <p>Servidor WEB: Apache2</p> <p>Base de Datos: María DB</p> <p>PHP: 8.0</p>

Descargamos la aplicación Rufus usando el navegador web en la siguiente URL: <https://rufus.ie/es/>, en la siguiente figura se muestra la aplicación en su modo portable.

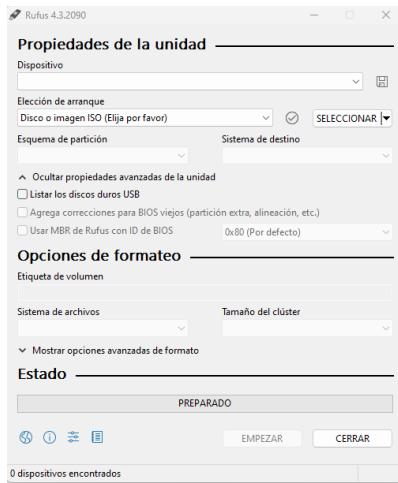


Figura 11. Aplicación Rufus en su modo portable.

Descargamos el sistema operativo Debian 12 en la siguiente URL:
<https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-12.2.0-amd64-netinst.iso>, como se muestra en la siguiente figura.



Figura 12. Página oficial de Debian

Procedemos a instalar Debian en nuestro servidor. Necesitaremos una USB mayor a 8 GB para guardar los archivos de instalación de la ISO de Debian. Abrimos la aplicación Rufus para crear una USB booteable donde se guardará la información de la imagen del sistema operativo Debian para posteriormente instalarlo en el servidor. (Para obtener más detalles de ingeniería, consulte el anexo A).



Figura 13. Configuración de Rufus para Bootear la ISO

Para establecer una conexión remota con el servidor, descargamos la aplicación PuTTY desde la URL: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Posteriormente, instalamos el paquete SSH en el servidor para permitir la conexión remota a través de PuTTY. Una vez instalado el paquete, abrimos la aplicación PuTTY y configuramos la conexión utilizando la dirección IP del servidor y un nombre de identificación específico.

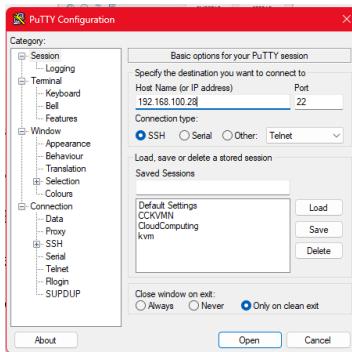


Figura 14. Configuración de la IP para conectar al servidor por SSH

Para facilitar la virtualización desde nuestros sistemas operativos, procedemos con la instalación del hipervisor QEMU en la terminal del servidor. Utilizaremos PuTTY para conectarnos de forma remota y ejecutar los comandos necesarios.

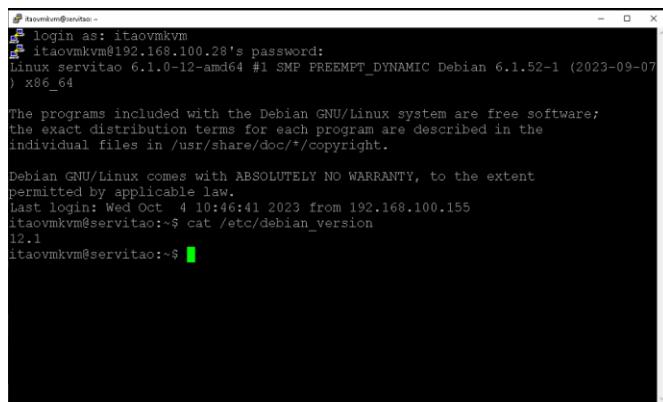


Figura 15. Terminal remota del servidor por SSH

Descargamos el repositorio de QEMU desde la URL: wget <https://download.qemu.org/qemu-8.1.0.tar.xz> y procedemos a descomprimirlo. Luego, llevamos a cabo la instalación según los pasos indicados.

Una vez instalado QEMU, configuramos la ruta de PATH para permitir la ejecución de QEME en cualquier ubicación dentro de la terminal.

Con estos comandos podremos instalar QEMU. (Para obtener más detalles de ingeniería, consulte el anexo A).

```
# tar Jvxf qemu-8.1.0.tar.xz

# ./configure--prefix=/opt/qemu-8 --target-list=i386-softmmu,x86_64-softmmu--enable-spice --
# enable-gtk

# make -j 16

# nano ~/.bashrc

# export PATH=$PATH:/root/qemu-8.1.0/build/qemu-bundle/opt/qemu-8/bin/

# source ~/.bashrc
```

3.3 Prueba

Comenzamos la prueba utilizando la aplicación Remote View que la descargaremos de la siguiente URL: <https://virt-manager.org/download.html> para visualizar el sistema operativo, considerando que QEMU está ejecutando el terminal y virtualizando el sistema operativo. Luego, implementaremos el protocolo SPICE para visualizar gráficamente el sistema operativo en otra computadora esto ambientado en cloud computing.



Figura 16. Remote Viewer con la ip necesaria

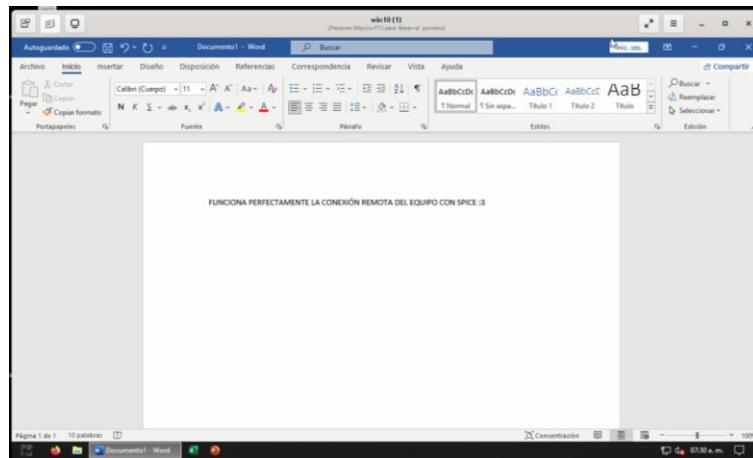


Figura 17. Ejecución de la maquina con Windows 10 usando SPICE y remote viewer

3.4 Integración

La elección entre Debian y Ubuntu depende en gran medida de las necesidades específicas del usuario y del entorno de trabajo. Si se prefiere un sistema más limpio y personalizable, Debian puede ser la opción ideal, ya que ofrece un entorno de trabajo que se adapta precisamente a lo que se necesita al instalar, sin procesos adicionales. Por otro lado, Ubuntu, con su enfoque más orientado al usuario, proporciona una experiencia más amigable y un mayor soporte para hardware y software.

La comparativa entre ambos sistemas en la tabla puede brindar una perspectiva más clara de las diferencias y similitudes fundamentales entre Debian y Ubuntu, lo que facilitará una toma de decisiones informada se muestra en la tabla 12.

Tabla 12. Comparativa entre Debian y Ubuntu

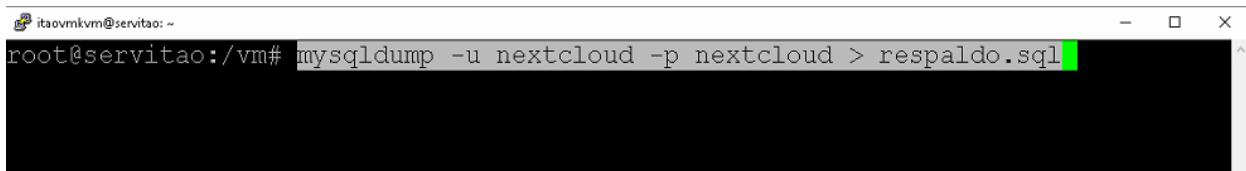
Características	Debian	Ubuntu
Enfoque en el servidor	Sí, se destaca por su estabilidad y limpieza en entornos de servidor.	También adecuado para servidores, pero con una orientación más amplia hacia el usuario.
Virtualización	Ofrece amplio soporte para virtualización con herramientas como QEMU y KVM.	Proporciona capacidades de virtualización similares, pero puede requerir una configuración más específica para ciertos casos.
Estabilidad	Conocido por su enfoque en la estabilidad y la confiabilidad, ideal para entornos de servidor críticos.	Ofrece un equilibrio entre la estabilidad y la disponibilidad de las últimas características, lo que puede implicar una menor estabilidad en comparación con Debian.
Personalización	Permite una mayor personalización, lo que resulta beneficioso en escenarios de virtualización y servidores especializados.	Si bien se puede personalizar, se inclina más hacia un enfoque de facilidad de uso, lo que puede limitar ciertas personalizaciones avanzadas.

Características	Debian	Ubuntu
Limpieza del sistema	Proporciona un sistema más liviano y limpio, permitiendo una administración más precisa de recursos en entornos de servidor.	Aunque es limpio, puede contener más aplicaciones y características adicionales en comparación con Debian, lo que podría requerir una administración adicional.

En esta fase, nos centramos en la migración de Nextcloud mediante Ubuntu a Debian, asegurándonos de configurar el entorno para permitir el uso del protocolo SPICE directamente desde una página web conectada con Nextcloud.

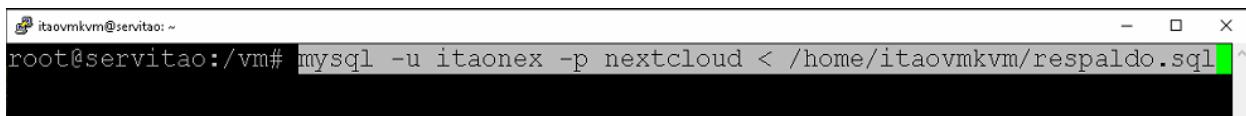
Nota: Es importante tener en cuenta que los comandos de Debian son compatibles con Ubuntu debido a que Ubuntu se deriva del sistema operativo Debian. Esto garantiza una considerable interoperabilidad entre ambos sistemas, lo que facilita el uso de comandos comunes en ambas plataformas.

Comenzamos la migración desde Ubuntu a Debian con la transferencia de la base de datos. Realizamos una copia de seguridad y la guardamos como un archivo .sql. Luego, en el entorno Debian, importamos el archivo .sql para restaurar la base de datos. Puedes encontrar los comandos específicos para estos pasos en la figura que se presenta a continuación. (Para obtener más detalles de ingeniería, consulte el anexo B).



```
itaovmkvm@servitao: ~
root@servitao:/vm# mysqldump -u nextcloud -p nextcloud > respaldo.sql
```

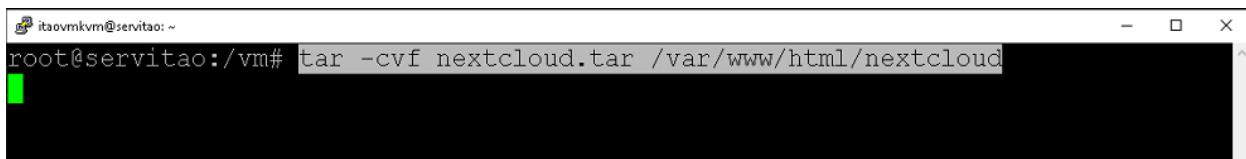
Figura 18. Comando para hacer una copia de seguridad a la base de datos.



```
itaovmkvm@servitao: ~
root@servitao:/vm# mysql -u itaonex -p nextcloud < /home/itaovmkvm/respaldo.sql
```

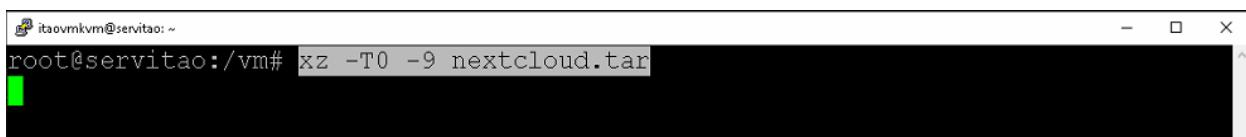
Figura 19. Comando para importar la base de datos a nuestro nuevo servidor.

El siguiente paso implica hacer una copia de seguridad del sistema Nextcloud. se empaqueta la carpeta completa en un archivo .tar y luego se comprime en un archivo. xz con un nivel de compresión de 9. Después de completar la compresión, puedes copiar la ubicación de la carpeta desde Ubuntu a Debian. La ruta de la carpeta en cuestión es /var/www/html/nextcloud/nextcloud, usando la aplicación FileZilla para mover los archivos correspondientes como se muestra en la siguiente figura. (Para obtener más detalles de ingeniería, consulte el anexo B).



```
itaovmkvm@servitao: ~
root@servitao:/vm# tar -cvf nextcloud.tar /var/www/html/nextcloud
```

Figura 20. Se empaqueta la carpeta en un .tar



```
itaovmkvm@servitao: ~
root@servitao:/vm# xz -T0 -9 nextcloud.tar
```

Figura 21. Comprimimos el archivo empaquetado en un .xz con nivel 9 para tener menos peso del archivo

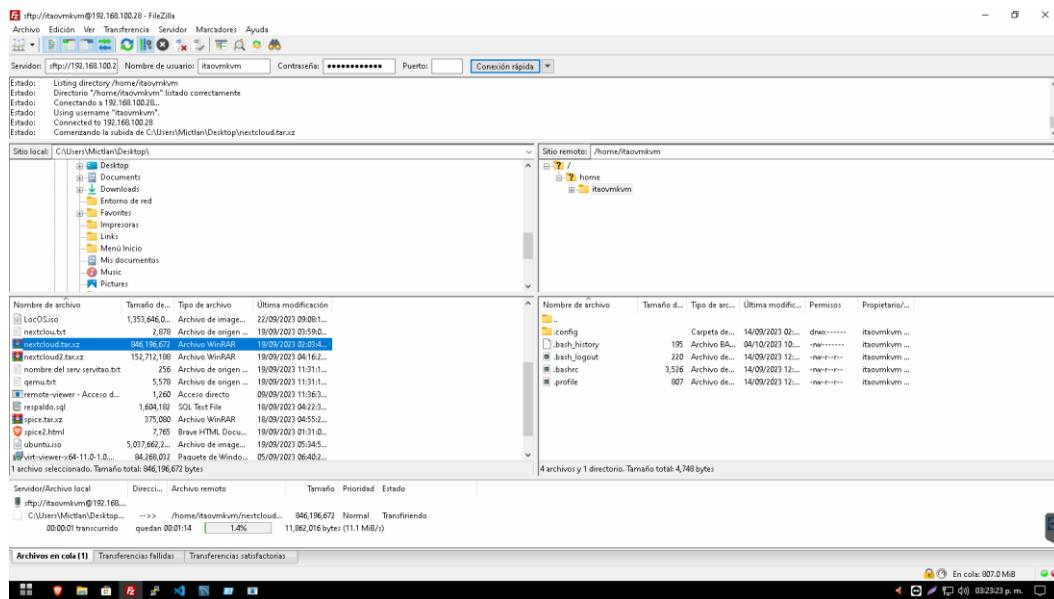


Figura 22. Usamos FileZilla para pasar los archivos correspondientes al servidor Debian.

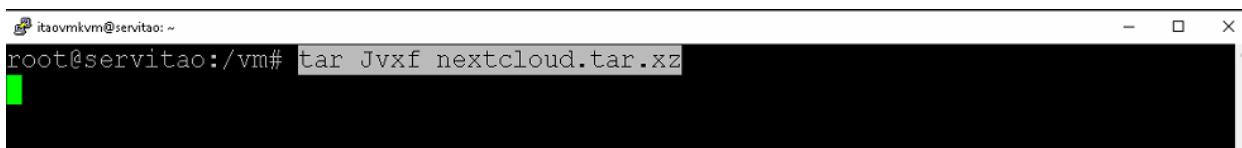


Figura 23. Descomprimimos el archivo

Tras la migración exitosa de Ubuntu a Debian, procederemos a activar el protocolo SPACE a través de Nextcloud. Esto implica la adición de una sección específica en los archivos, haciendo uso del repositorio SPACE. A continuación, se proporcionará un fragmento del código relevante, junto con una descripción de cómo funciona una vez activado en una página web mediante Nextcloud. (Para obtener más detalles de ingeniería, consulte el anexo B).

Instituto Tecnológico de Álvaro Obregón

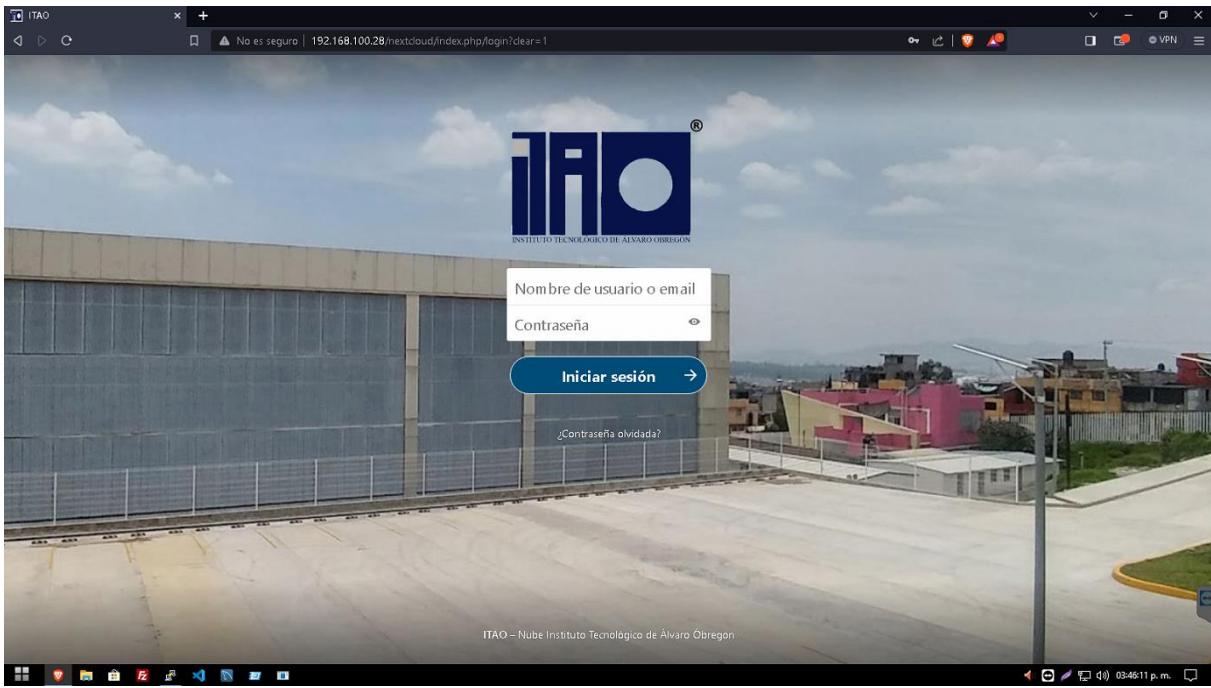


Figura 24. Inicio de sesión de nextcloud

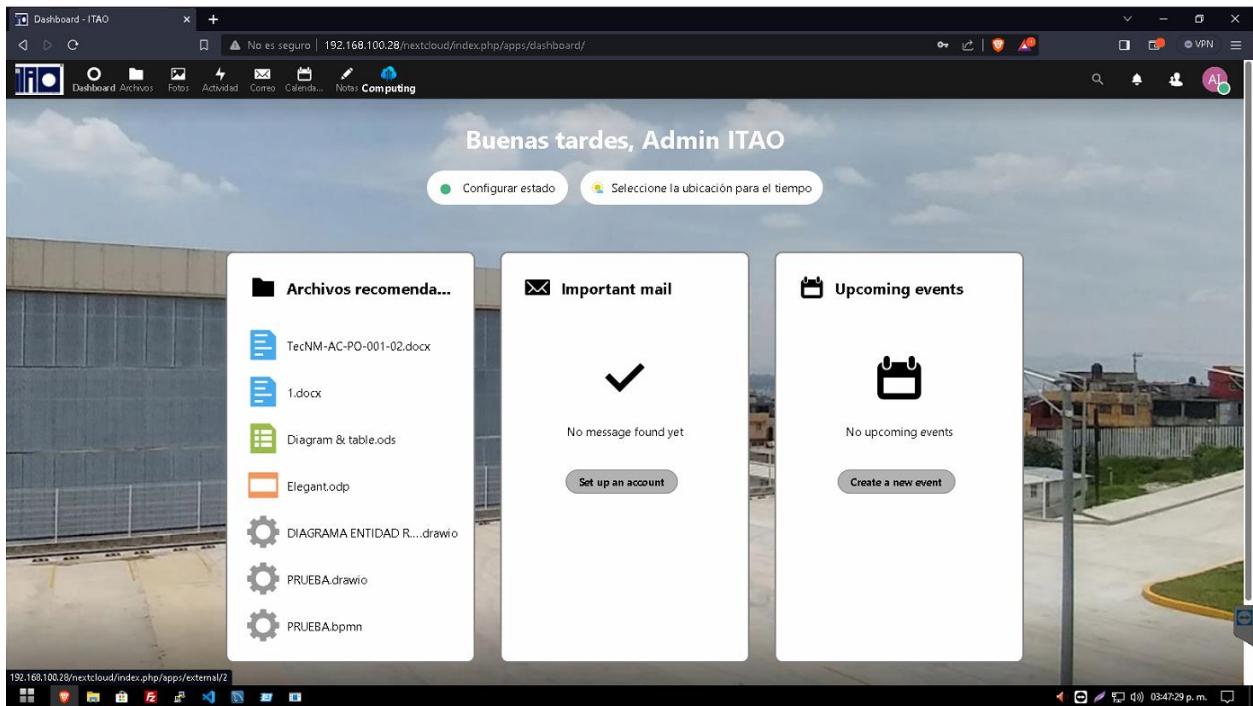
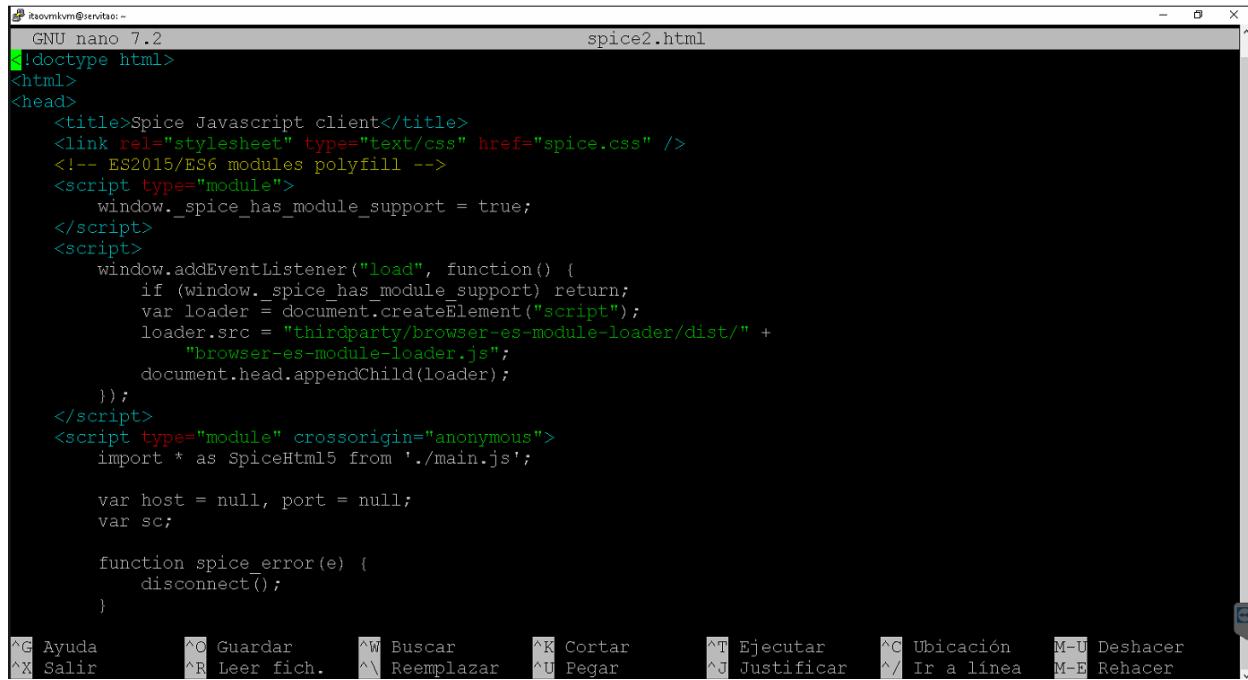


Figura 25. Página de Inicio de Nextcloud

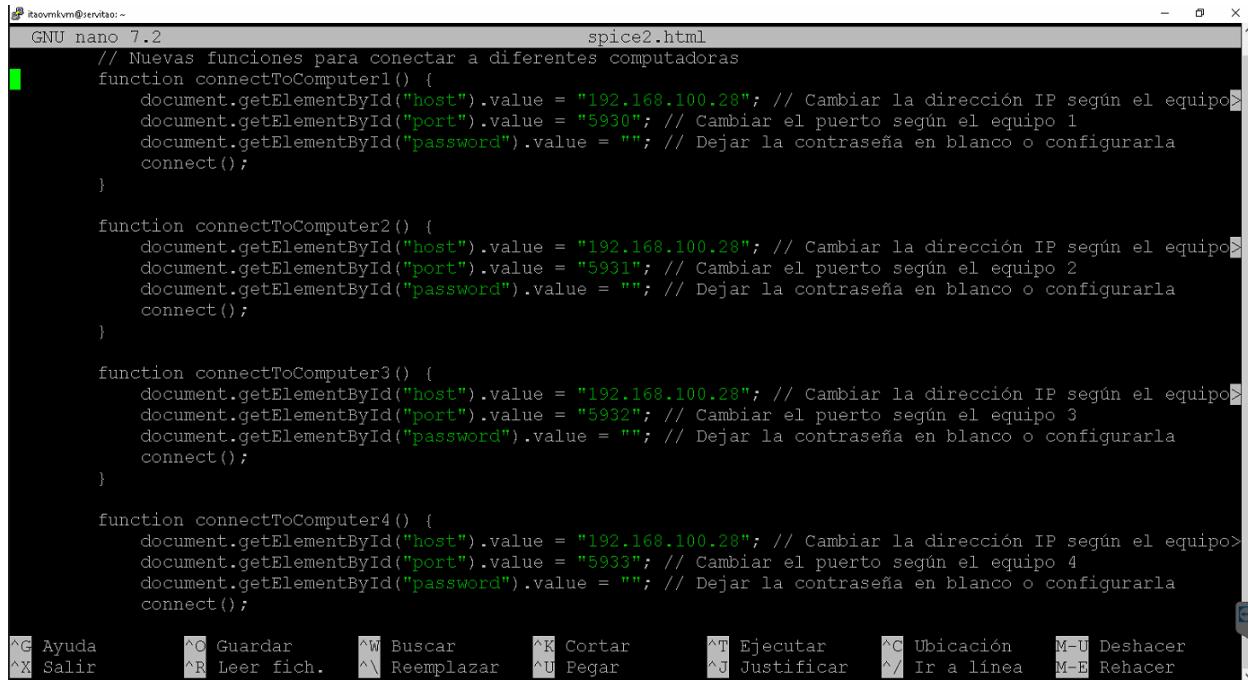


```
iteavmkvm@servitao: ~
GNU nano 7.2
spice2.html
<!doctype html>
<html>
<head>
<title>Spice Javascript client</title>
<link rel="stylesheet" type="text/css" href="spice.css" />
<!-- ES2015/ES6 modules polyfill -->
<script type="module">
    window._spice_has_module_support = true;
</script>
<script>
    window.addEventListener("load", function() {
        if (window._spice_has_module_support) return;
        var loader = document.createElement("script");
        loader.src = "thirdparty/browser-es-module-loader/dist/" +
            "browser-es-module-loader.js";
        document.head.appendChild(loader);
    });
</script>
<script type="module" crossorigin="anonymous">
    import * as SpiceHtml5 from './main.js';

    var host = null, port = null;
    var sc;

    function spice_error(e) {
        disconnect();
    }
</script>
^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación      M-U Deshacer
^X Salir      ^R Leer fich.  ^\ Reemplazar   ^U Pegar       ^J Justificar   ^/ Ir a linea   M-E Rehacer
```

Figura 26. Código para ejecutar el entorno virtual por el protocolo SPICE



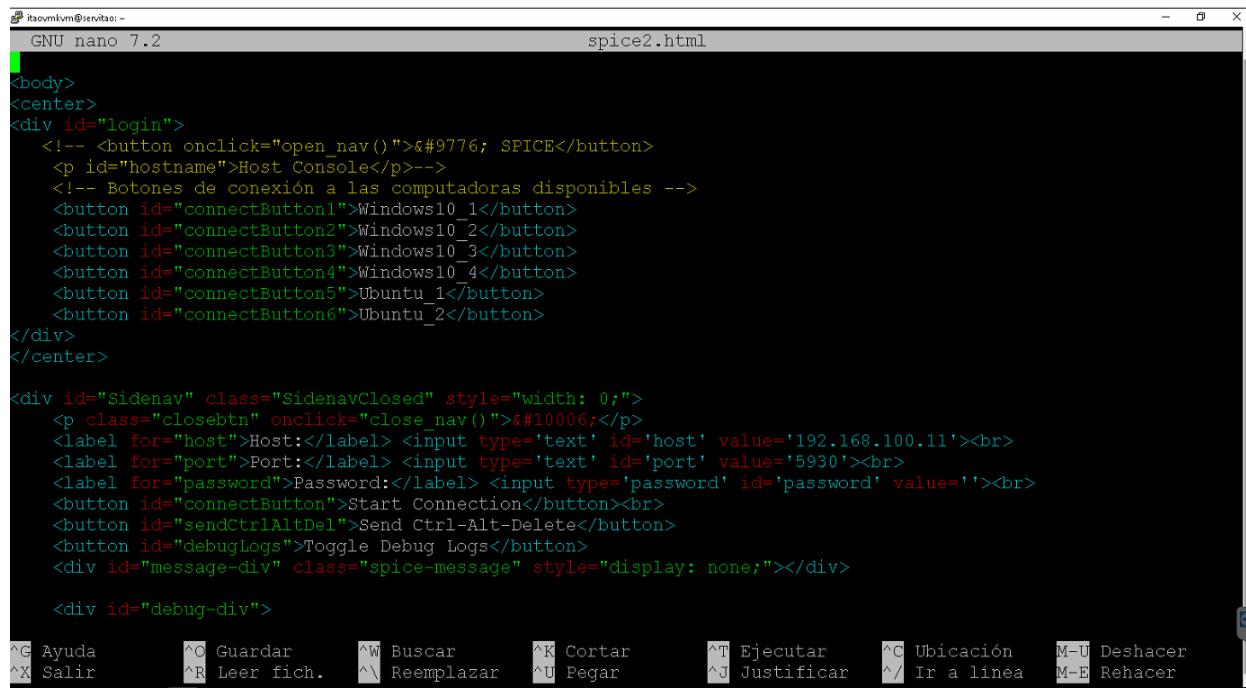
```
iteavmkvm@servitao: ~
GNU nano 7.2
spice2.html
// Nuevas funciones para conectar a diferentes computadoras
function connectToComputer1() {
    document.getElementById("host").value = "192.168.100.28"; // Cambiar la dirección IP según el equipo 1
    document.getElementById("port").value = "5930"; // Cambiar el puerto según el equipo 1
    document.getElementById("password").value = ""; // Dejar la contraseña en blanco o configurarla
    connect();
}

function connectToComputer2() {
    document.getElementById("host").value = "192.168.100.28"; // Cambiar la dirección IP según el equipo 2
    document.getElementById("port").value = "5931"; // Cambiar el puerto según el equipo 2
    document.getElementById("password").value = ""; // Dejar la contraseña en blanco o configurarla
    connect();
}

function connectToComputer3() {
    document.getElementById("host").value = "192.168.100.28"; // Cambiar la dirección IP según el equipo 3
    document.getElementById("port").value = "5932"; // Cambiar el puerto según el equipo 3
    document.getElementById("password").value = ""; // Dejar la contraseña en blanco o configurarla
    connect();
}

function connectToComputer4() {
    document.getElementById("host").value = "192.168.100.28"; // Cambiar la dirección IP según el equipo 4
    document.getElementById("port").value = "5933"; // Cambiar el puerto según el equipo 4
    document.getElementById("password").value = ""; // Dejar la contraseña en blanco o configurarla
    connect();
}
^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación      M-U Deshacer
^X Salir      ^R Leer fich.  ^\ Reemplazar   ^U Pegar       ^J Justificar   ^/ Ir a linea   M-E Rehacer
```

Figura 27. Código en JS para conectar las ip con los puertos adecuados para el protocolo SPICE



```

Itavm1vm@servidor: ~
GNU nano 7.2
spice2.html

<body>
<center>
<div id="login">
  <!-- <button onclick="open_nav()">SPICE</button>
  <p id="hostname">Host Console</p>-->
  <!-- Botones de conexión a las computadoras disponibles -->
  <button id="connectButton1">Windows10_1</button>
  <button id="connectButton2">Windows10_2</button>
  <button id="connectButton3">Windows10_3</button>
  <button id="connectButton4">Windows10_4</button>
  <button id="connectButton5">Ubuntu_1</button>
  <button id="connectButton6">Ubuntu_2</button>
</div>
</center>

<div id="Sidenav" class="SidenavClosed" style="width: 0;">
  <p class="closebtn" onclick="close_nav()">&#10006;</p>
  <label for="host">Host:</label> <input type='text' id='host' value='192.168.100.11'><br>
  <label for="port">Port:</label> <input type='text' id='port' value='5930'><br>
  <label for="password">Password:</label> <input type='password' id='password' value=' '><br>
  <button id="connectButton">Start Connection</button><br>
  <button id="sendCtrlAltDel">Send Ctrl-Alt-Delete</button>
  <button id="debugLogs">Toggle Debug Logs</button>
  <div id="message-div" class="spice-message" style="display: none;"></div>
</div>
<div id="debug-div">

  ^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación      M-U Deshacer
  ^X Salir      ^R Leer fich.   ^\ Reemplazar   ^U Pegar       ^J Justificar   ^/ Ir a linea   M-E Rehacer

```

Figura 28. Código en HTML para el entorno visual.

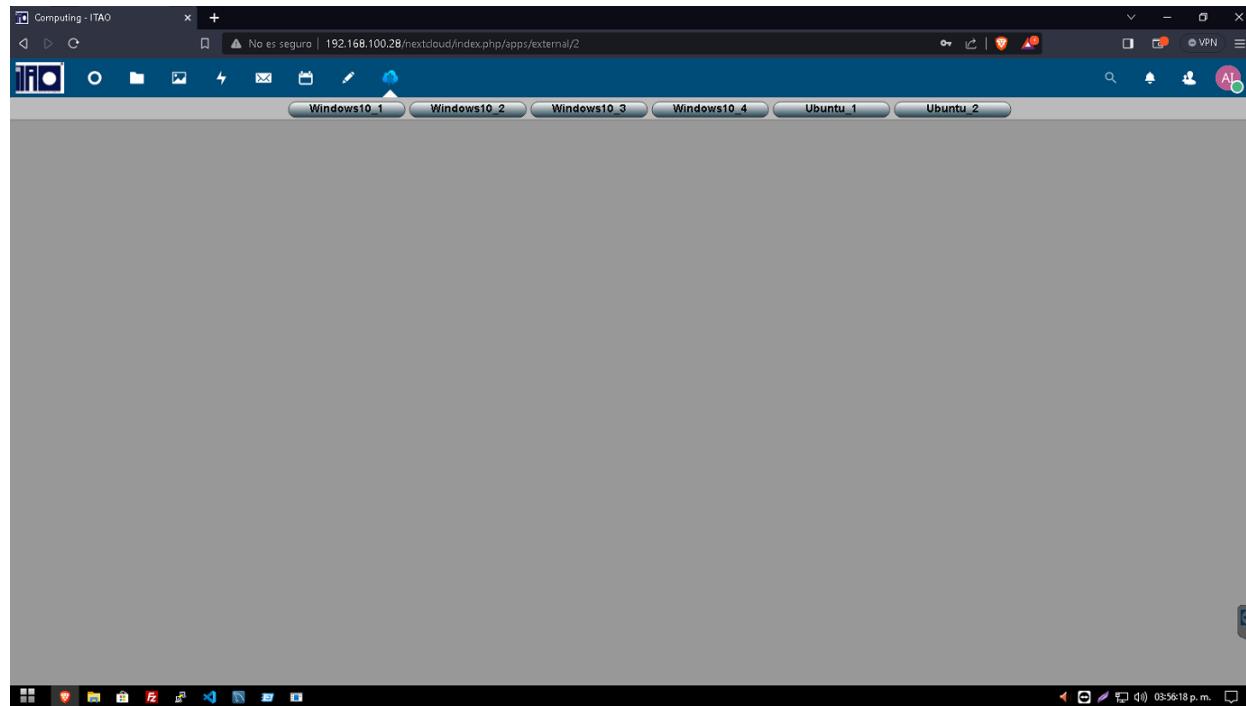


Figura 29. Entorno para ejecutar nuestro sistema operativo

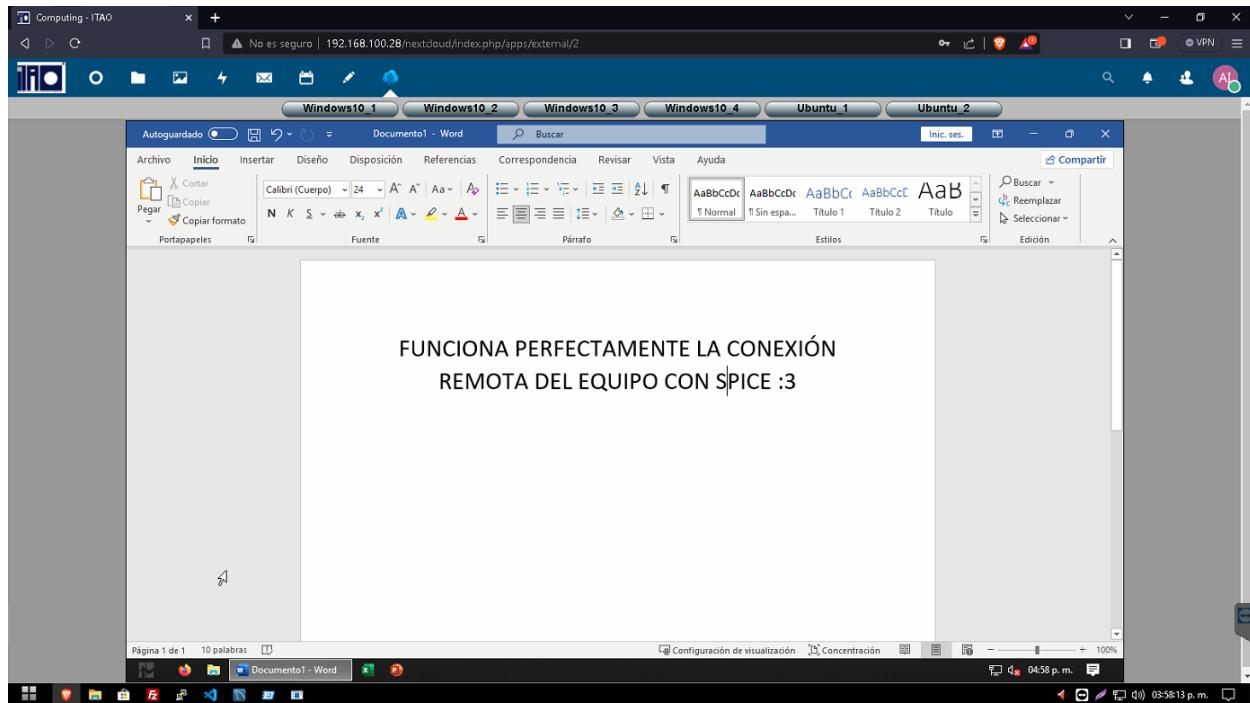


Figura 30. Ejecución de Word mediante Cloud Computing

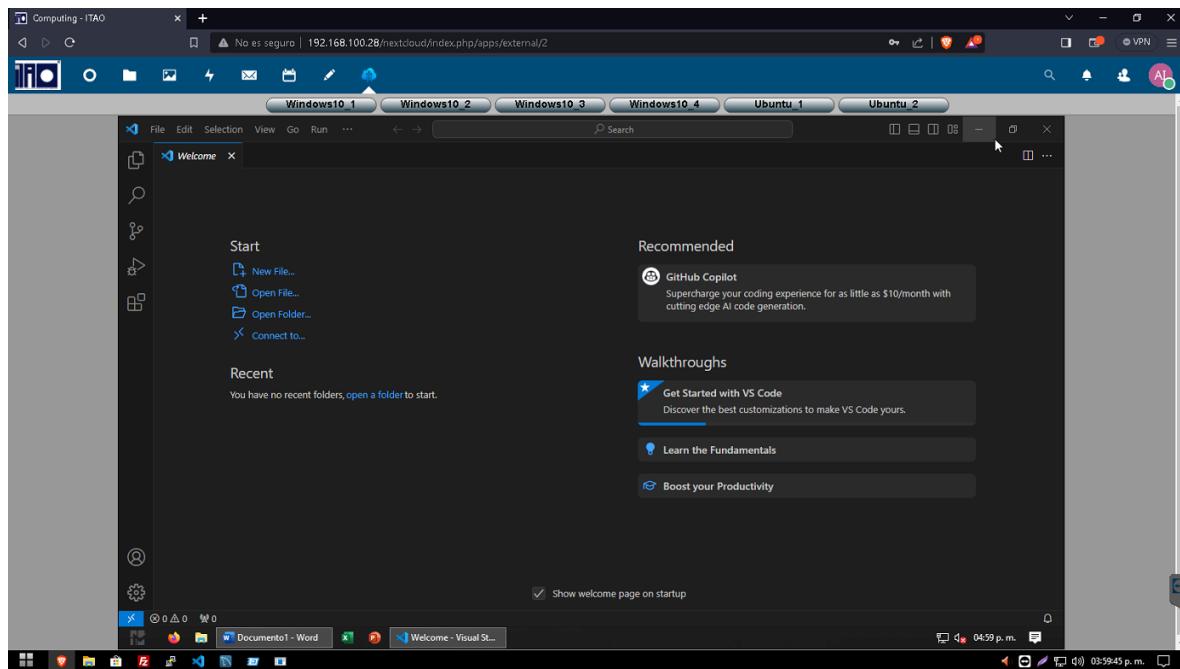


Figura 31. Ejecución de Visual Code mediante Cloud Computing

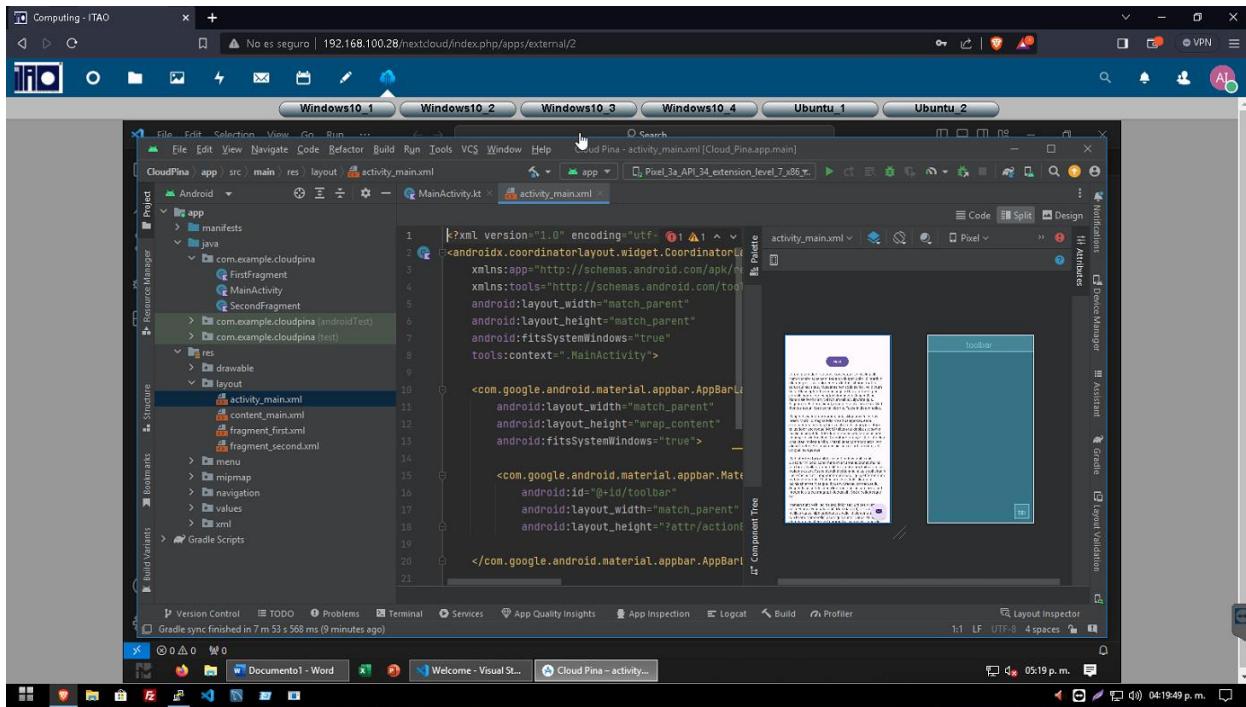


Figura 32. Ejecución de Android Studio en modo Cloud Computing

3.5 Despliegue

Ahora que el proyecto de virtualización de cómputo en la nube en el servidor está plenamente funcional, avanzaremos con el despliegue del contenido en la red.

Garantizar el funcionamiento ininterrumpido de los equipos, permitiéndoles acceder a diversas aplicaciones. Durante este proceso, buscamos alcanzar un punto de estabilidad en el que las máquinas puedan ejecutar de manera fluida una variedad de aplicaciones, incluyendo Word, Visual Code, Android Studio y navegadores, entre otras herramientas.

Este enfoque nos ayudará a evaluar la capacidad del servidor bajo una carga intensa y determinar su rendimiento óptimo.

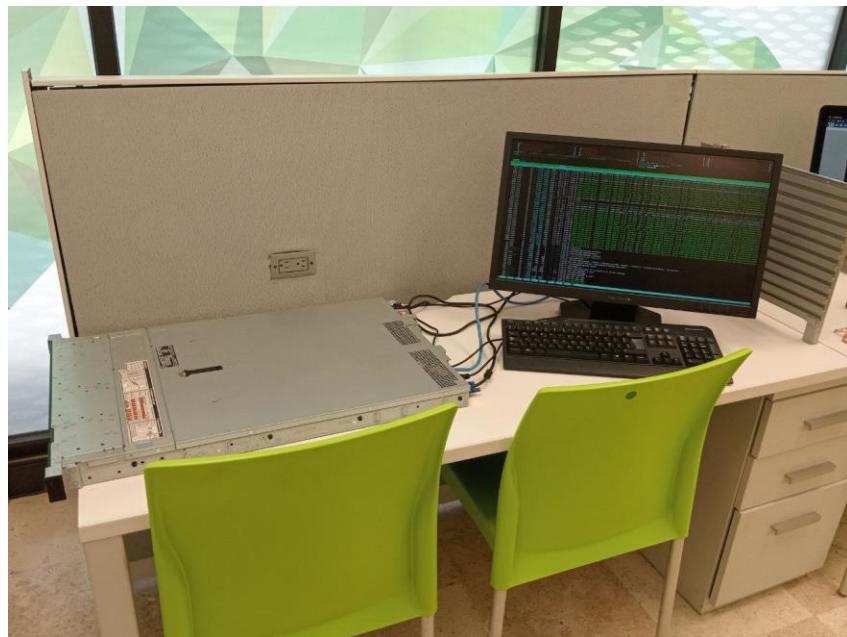


Figura 33. Servidor conectado para el despliegue.

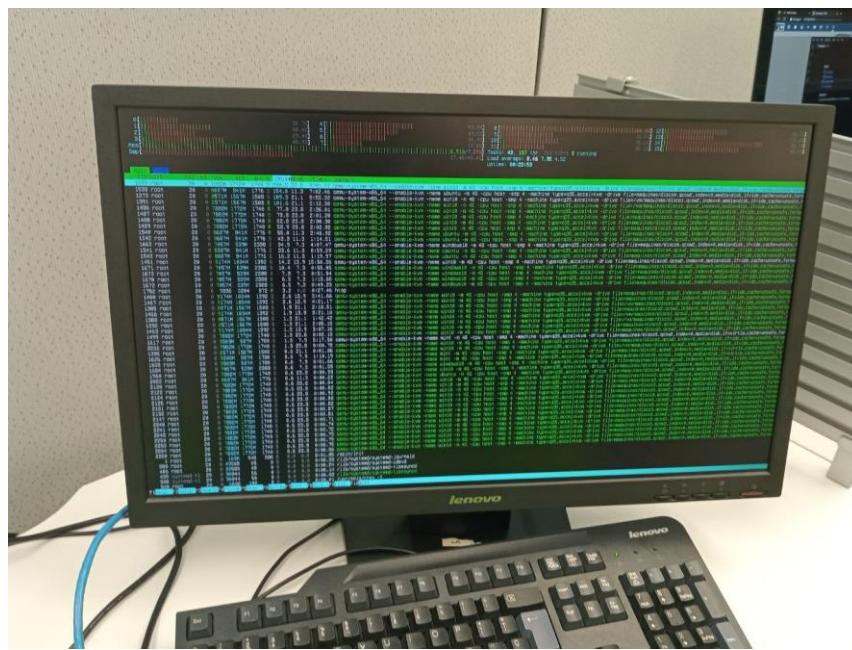


Figura 34. Monitoreo del servidor para ver su estabilidad con el despliegue del servicio SaaS.



Figura 35. 6 equipos mostrando la virtualización desde nextcloud.

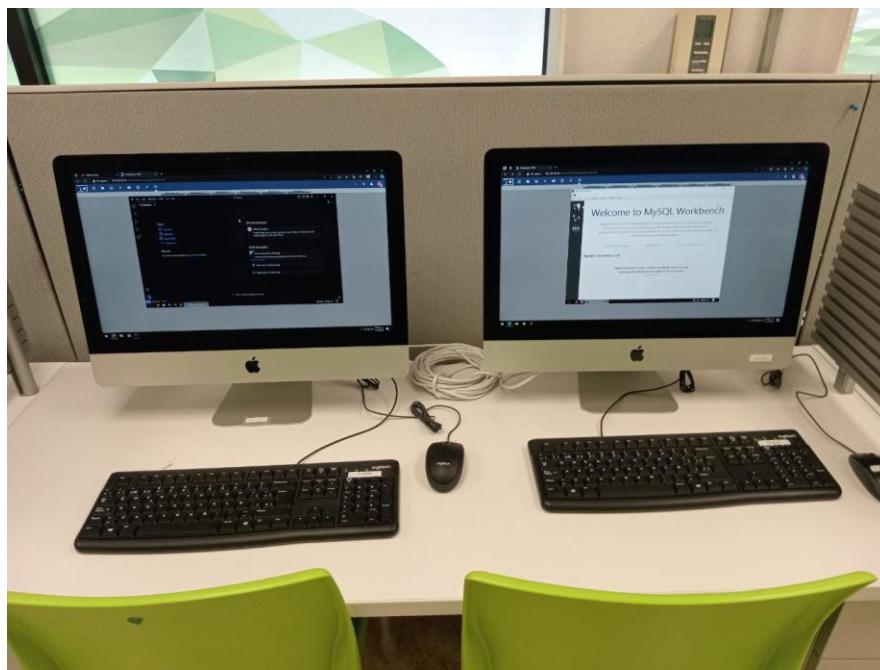


Figura 36. Se muestra la aplicación visual code y mysql workbench

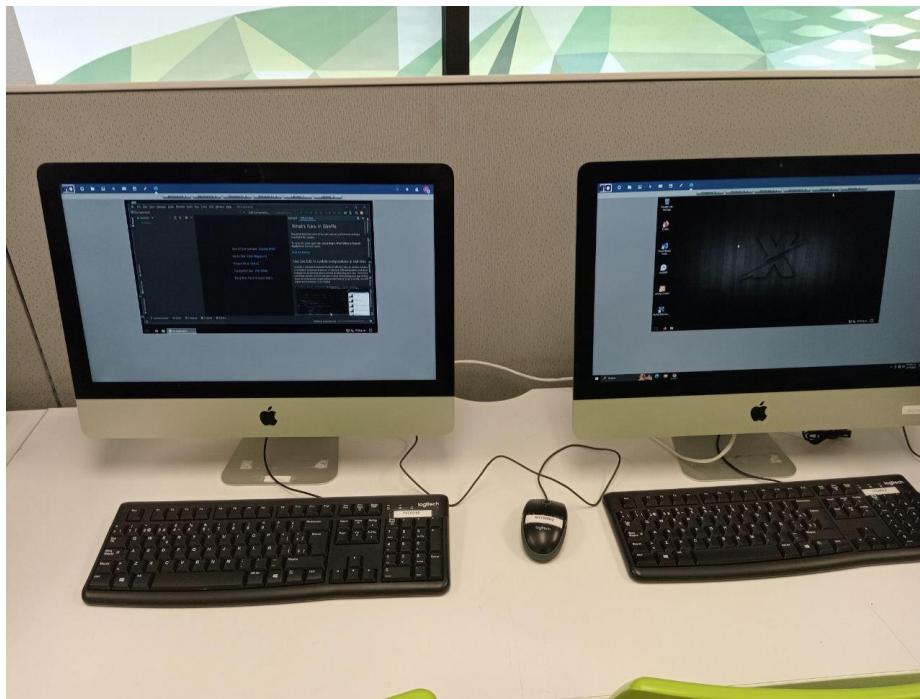


Figura 37. Se muestra la aplicación Android studio y el Windows sin ejecutar nada

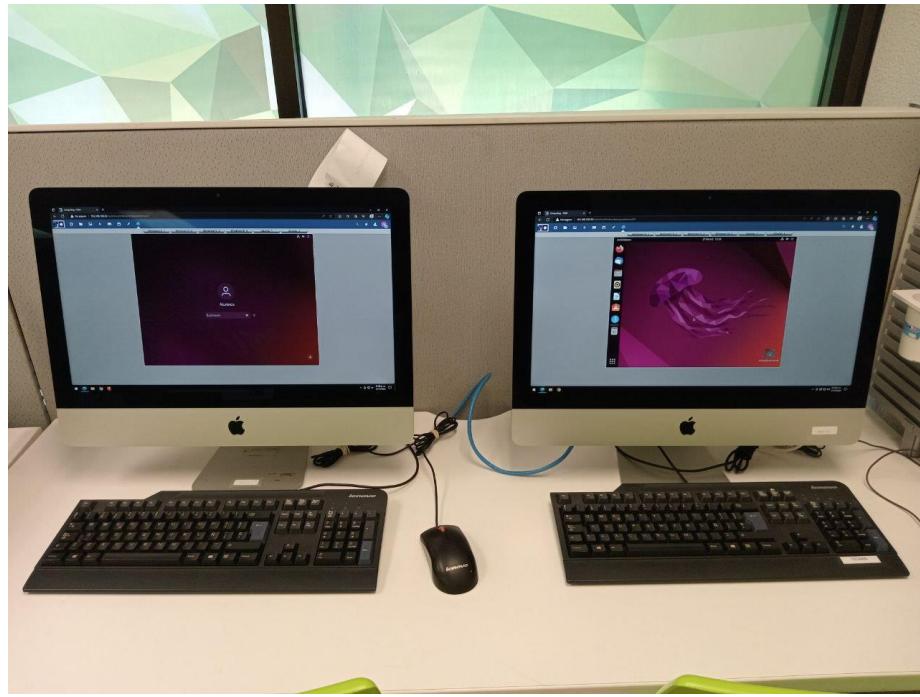
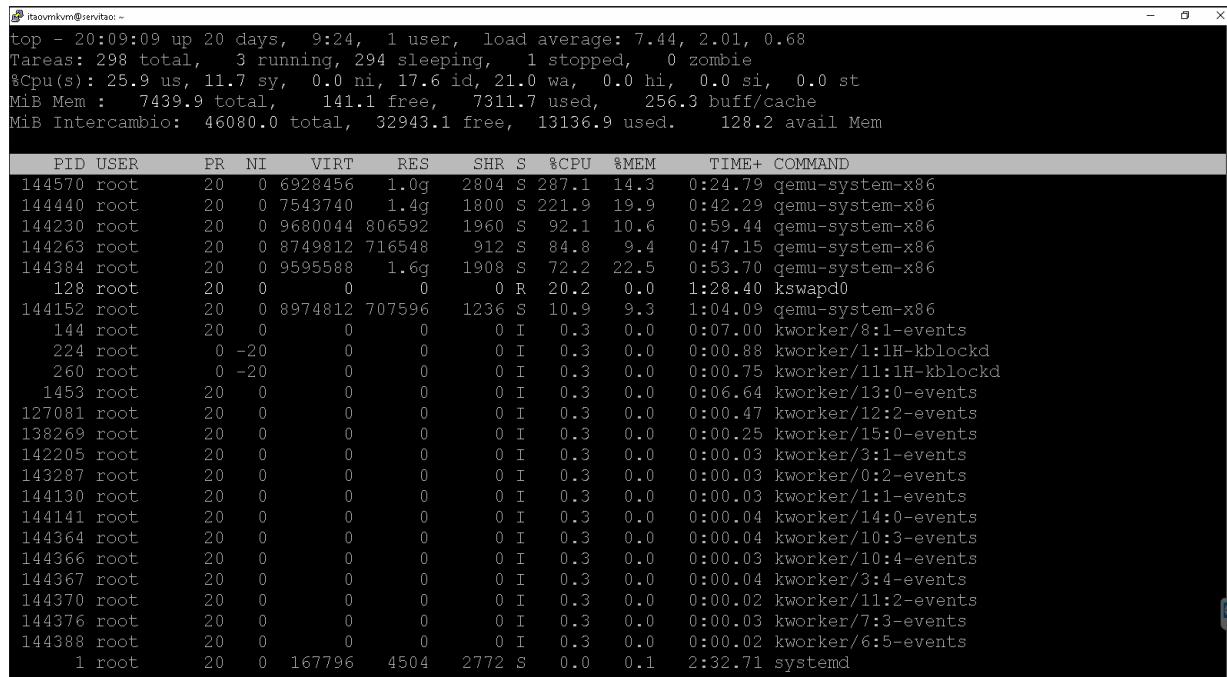


Figura 38. Se muestra que también se puede ejecutar Linux como Ubuntu

3.6 Monitoreo

En esta etapa final, evaluaremos minuciosamente la funcionalidad del servidor utilizando herramientas específicas para monitorear el rendimiento del sistema. Para este propósito, aprovecharemos dos herramientas clave. En primer lugar, utilizaremos TOP para supervisar el consumo de la CPU, la memoria y otros procesos en tiempo real. Además, para un análisis más detallado y visual del rendimiento, emplearemos HTOP, que nos proporciona una representación gráfica más clara del uso del CPU y de la memoria. Con HTOP, podremos examinar individualmente el rendimiento de cada núcleo e hilo, así como determinar el uso de la memoria RAM y el porcentaje de uso del área de intercambio (swap). Esto nos permitirá identificar cualquier proceso invasivo que pueda afectar el rendimiento del sistema y tomar medidas adecuadas para optimizar su funcionamiento.



```

top - 20:09:09 up 20 days,  9:24,  1 user,  load average: 7.44, 2.01, 0.68
Tareas: 298 total,   3 running, 294 sleeping,   1 stopped,   0 zombie
%Cpu(s): 25.9 us, 11.7 sy,  0.0 ni, 17.6 id, 21.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 7439.9 total,   141.1 free,   7311.7 used,    256.3 buff/cache
MiB Intercambio: 46080.0 total,  32943.1 free,  13136.9 used,   128.2 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
144570 root      20   0 6928456  1.0g  2804 S 287.1 14.3  0:24.79 qemu-system-x86
144440 root      20   0 7543740  1.4g  1800 S 221.9 19.9  0:42.29 qemu-system-x86
144230 root      20   0 9680044 806592  1960 S 92.1 10.6  0:59.44 qemu-system-x86
144263 root      20   0 8749812 716548  912 S 84.8  9.4  0:47.15 qemu-system-x86
144384 root      20   0 9595588  1.6g  1908 S 72.2 22.5  0:53.70 qemu-system-x86
  128 root      20   0     0     0     0 R 20.2  0.0  1:28.40 kswapd0
144152 root      20   0 8974812 707596  1236 S 10.9  9.3  1:04.09 qemu-system-x86
  144 root      20   0     0     0     0 I  0.3  0.0  0:07.00 kworker/8:1-events
  224 root      0 -20     0     0     0 I  0.3  0.0  0:00.88 kworker/1:1H-kblockd
  260 root      0 -20     0     0     0 I  0.3  0.0  0:00.75 kworker/11:1H-kblockd
  1453 root      20   0     0     0     0 I  0.3  0.0  0:06.64 kworker/13:0-events
127081 root      20   0     0     0     0 I  0.3  0.0  0:00.47 kworker/12:2-events
138269 root      20   0     0     0     0 I  0.3  0.0  0:00.25 kworker/15:0-events
142205 root      20   0     0     0     0 I  0.3  0.0  0:00.03 kworker/3:1-events
143287 root      20   0     0     0     0 I  0.3  0.0  0:00.03 kworker/0:2-events
144130 root      20   0     0     0     0 I  0.3  0.0  0:00.03 kworker/1:1-events
144141 root      20   0     0     0     0 I  0.3  0.0  0:00.04 kworker/14:0-events
144364 root      20   0     0     0     0 I  0.3  0.0  0:00.04 kworker/10:3-events
144366 root      20   0     0     0     0 I  0.3  0.0  0:00.03 kworker/10:4-events
144367 root      20   0     0     0     0 I  0.3  0.0  0:00.04 kworker/3:4-events
144370 root      20   0     0     0     0 I  0.3  0.0  0:00.02 kworker/11:2-events
144376 root      20   0     0     0     0 I  0.3  0.0  0:00.03 kworker/7:3-events
144388 root      20   0     0     0     0 I  0.3  0.0  0:00.02 kworker/6:5-events
    1 root      20   0 167796  4504  2772 S  0.0  0.1  2:32.71 systemd

```

Figura 39. Aplicación de monitoreo de procesos y rendimiento llamado TOP

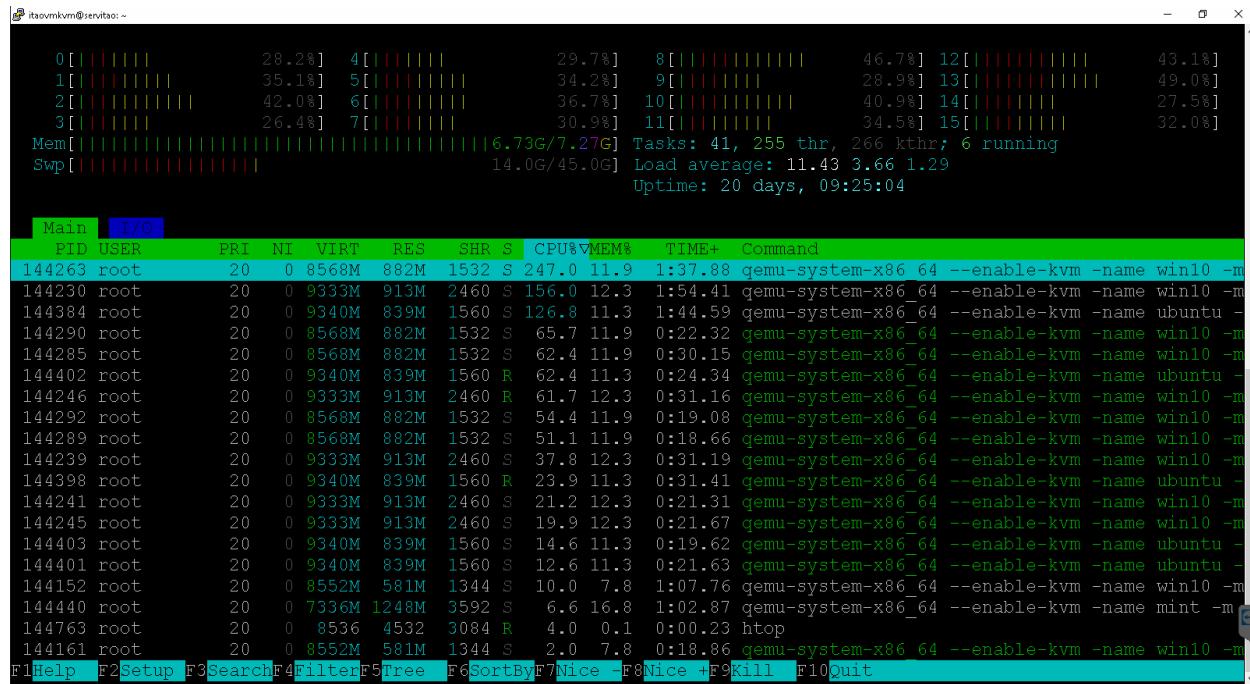
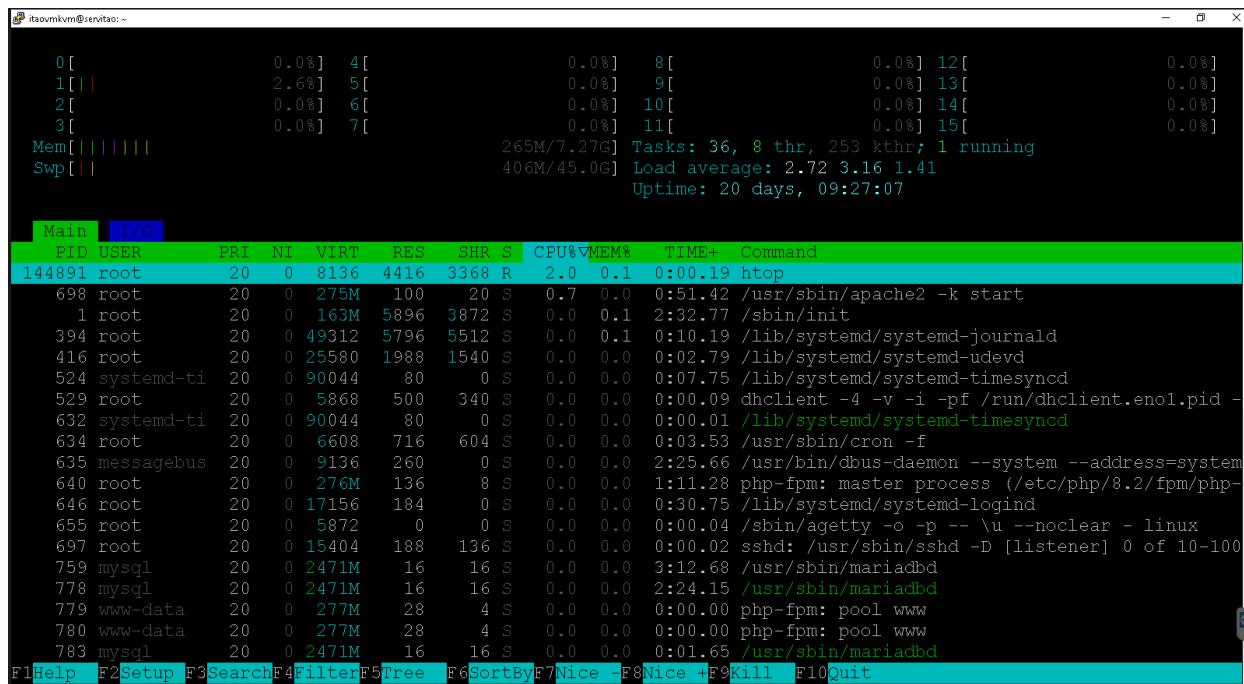


Figura 40. Aplicación de monitoreo de procesos y rendimiento llamado HTOP con entorno más grafico

```
root@servitao:/vm# kill 144384
root@servitao:/vm# kill 144230
root@servitao:/vm# kill 144263
root@servitao:/vm# kill 144570
root@servitao:/vm# kill 144152
root@servitao:/vm# kill 144440
root@servitao:/vm#
```

Figura 41. Si desconocemos un proceso, debemos asegurarnos de terminarlo para evitar cualquier posible infiltración.

En la figura anterior se mostraron los procesos eliminados con el comando ‘kill’, y ahora se ve que dejaron de consumir recursos del servidor.


Figura 42. Procesos eliminados

Conclusión

En este arduo trabajo, se llevó a cabo la virtualización mediante un hipervisor de tipo 1 utilizando QEMU y KVM, lo cual resulta altamente factible para proporcionar a los estudiantes un entorno de computación óptimo sin la necesidad de costosos equipos individuales en un laboratorio. Además, se implementó Nextcloud para extender el protocolo SPICE, lo que facilita la virtualización basada en la web, haciendo que el uso de las máquinas sea mucho más accesible y amigable para los usuarios. Esto permite un control exhaustivo tanto de las máquinas virtuales como de los programas, asignando una máquina virtual a cada estudiante. De esta manera, se facilita la demanda de recursos de cómputo, permitiendo ejecutar programas que requieran un alto rendimiento. Se puede configurar cada máquina virtual de forma específica, lo que conlleva a una mayor estabilidad, control y eficiencia, posibilitando que los estudiantes entreguen sus proyectos a tiempo sin obstáculos relacionados con la potencia del equipo o la disponibilidad de recursos.

Recomendaciones

- **Explorar el uso de Bash:** Excelente punto. Aprender Bash es fundamental, ya que te permite automatizar tareas y realizar operaciones avanzadas en sistemas Unix/Linux.
- **Selección del sistema operativo:** Destacar las ventajas específicas de Debian es útil. Además, podrías sugerir que se consideren otros sistemas operativos basados en las necesidades específicas del proyecto, como la estabilidad, la seguridad o el soporte para hardware específico.

- **Explorar herramientas adicionales:** Además de Nextcloud y el protocolo Space, podrías ampliar las recomendaciones con otras herramientas o servicios compatibles con Cloud Computing que faciliten el despliegue y la gestión de aplicaciones, tanto en entornos web como en entornos de escritorio remoto.

Experiencia profesional adquirida.

Uso de Bash

Instalación de servidores y manejo de redes

Implementación remota en entornos Cloud Computing

Uso de Protocolos de Red

Uso de APIs



Referencias

Amazon Web Service. (28 de 09 de 2023). AWS. Obtenido de AWS: https://docs.aws.amazon.com/es_es/whitepapers/latest/aws-overview/types-of-cloud-computing.html

Apache. (30 de 09 de 2023). apache. Obtenido de apache: https://httpd.apache.org/docs/trunk/es/new_features_2_2.html

CISCO. (30 de 09 de 2023). cisco. Obtenido de cisco: <https://www.cisco.com/en/US/docs/security/vpn5000/manager/reference/guide/IProutg.pdf>

datascientest. (17 de 10 de 2023). datascientest. Obtenido de datascientest: <https://datascientest.com/es/devops-que-es-definicion-ventajas-cursos>

Debian. (30 de 09 de 2023). debian. Obtenido de debian: <https://www.debian.org/releases/stable/i386/release-notes/ch-whats-new.es.html>

DELL. (30 de 09 de 2023). DELL. Obtenido de DELL: <https://qrl.dell.com/FTYWJ93>

FileZilla. (30 de 09 de 2023). FileZilla. Obtenido de FileZilla: <https://filezilla-project.org/changelog.php?page=71&type=1&changes=25>

Hostiger. (30 de 09 de 2023). Hostiger. Obtenido de Hostiger: <https://www.hostinger.es/tutoriales/bash-script-linux>

IBM. (28 de 09 de 2023). IBM. Obtenido de IBM: <https://www.ibm.com/mx-es/topics/hypervisors>

INOS. (30 de 09 de 2023). inos. Obtenido de inos: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-archivo-iso/>

Nextcloud. (30 de 09 de 2023). Nextcloud. Obtenido de Nextcloud: <https://nextcloud.com/es/ingenieria/>

Oracle. (28 de 09 de 2023). OCI. Obtenido de OCI: <https://www.oracle.com/mx/cloud/what-is-cloud-computing/>

QEMU. (30 de 09 de 2023). qemu. Obtenido de qemu: <https://www.qemu.org/docs/master/system/introduction.html>

Red Hat. (30 de 09 de 2023). red hat. Obtenido de red hat: <https://www.redhat.com/es/topics/virtualization/what-is-KVM>

Red Hat. (28 de 09 de 2023). Red Hat. Obtenido de Red Hat: <https://www.redhat.com/es/topics/cloud-computing/public-cloud-vs-private-cloud-and-hybrid-cloud>

Red Hat. (28 de 09 de 2023). *Red Hat.* Obtenido de Red Hat:
<https://www.redhat.com/es/topics/cloud-computing/what-are-cloud-services>

Stackscale. (28 de 09 de 2023). *stackscale.* Obtenido de stackscale:
<https://www.stackscale.com/es/blog/modelos-de-servicio-cloud/>

Stackscale. (28 de 09 de 2023). *stackscale.* Obtenido de stackscale:
<https://www.stackscale.com/es/blog/hipervisores/>

Stackscale. (28 de 09 de 2023). *Stackscale.* Obtenido de Stackscale:
<https://www.stackscale.com/es/blog/tipos-cloud-privado-publico-hibrido/>

Wordpress. (30 de 09 de 2023). *wordpress.* Obtenido de wordpress:
<https://brayangp.files.wordpress.com/2015/02/que-es-putty.pdf>

ANEXO A

Descargamos la aplicación Rufus usando el navegador web en la siguiente URL:

<https://rufus.ie/es/>, en la siguiente figura se muestra la aplicación en su modo portable.

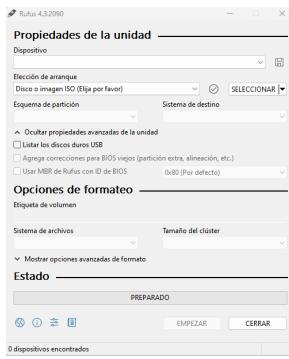


Figura 43. Programa Rufus ejecutando.

Descargamos el sistema operativo Debian 12 en la siguiente URL:

<https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-12.2.0-amd64-netinst.iso>,

como se muestra en la siguiente figura.

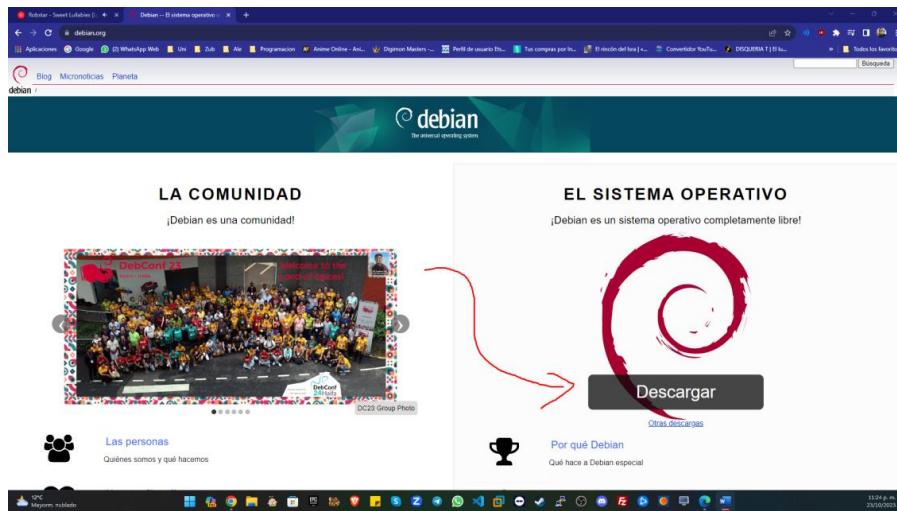


Figura 44. Página Web Oficial para descargar Debian 12

Procedemos a instalar Debian en nuestro servidor. Necesitaremos una USB mayor a 8 GB para guardar los archivos de instalación de la ISO de Debian. Abrimos la aplicación Rufus para crear una USB booteable donde se guardará la información de la imagen del sistema operativo Debian para posteriormente instalarlo en el servidor.



Figura 45. Aplicación Rufus lista para Boolear Debian en una USB

Para iniciar el arranque desde la USB en el servidor, primero enciende el sistema y presiona la tecla F2 o F1, dependiendo del modelo del servidor, para acceder a la configuración del BIOS. Una vez dentro, selecciona la opción de inicio desde USB. Esto te permite acceder a la siguiente pantalla como se ve en la siguiente figura.

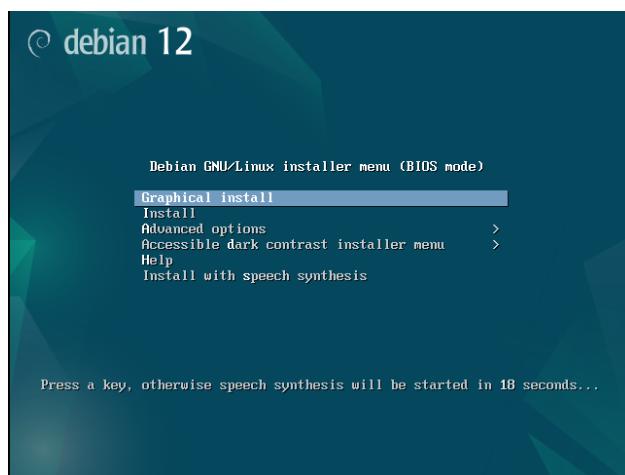


Figura 46. Menú de instalación gráfica o en terminal

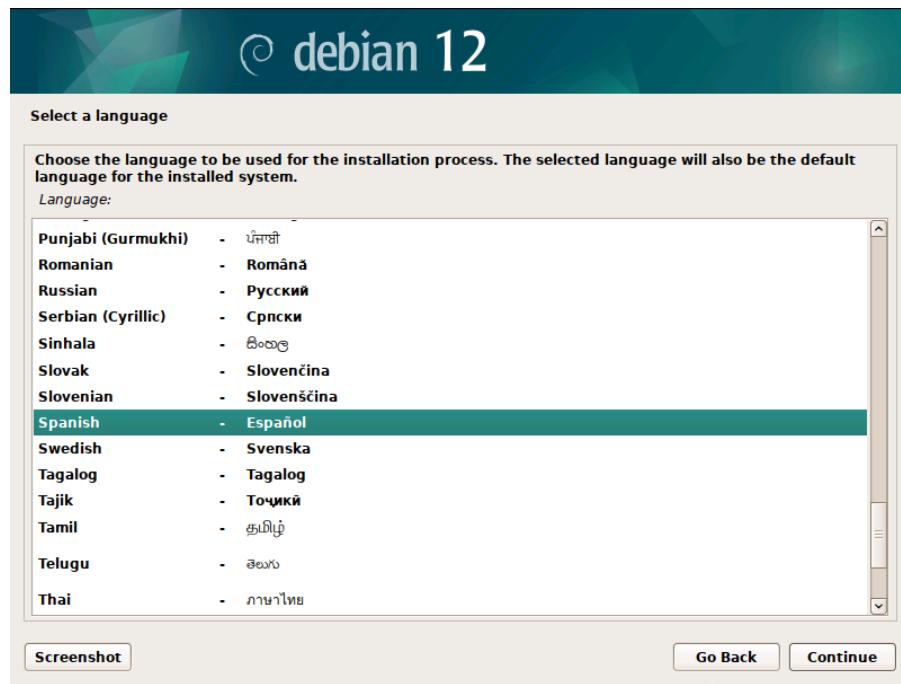


Figura 47. Seleccionamos el idioma del Sistema operativo

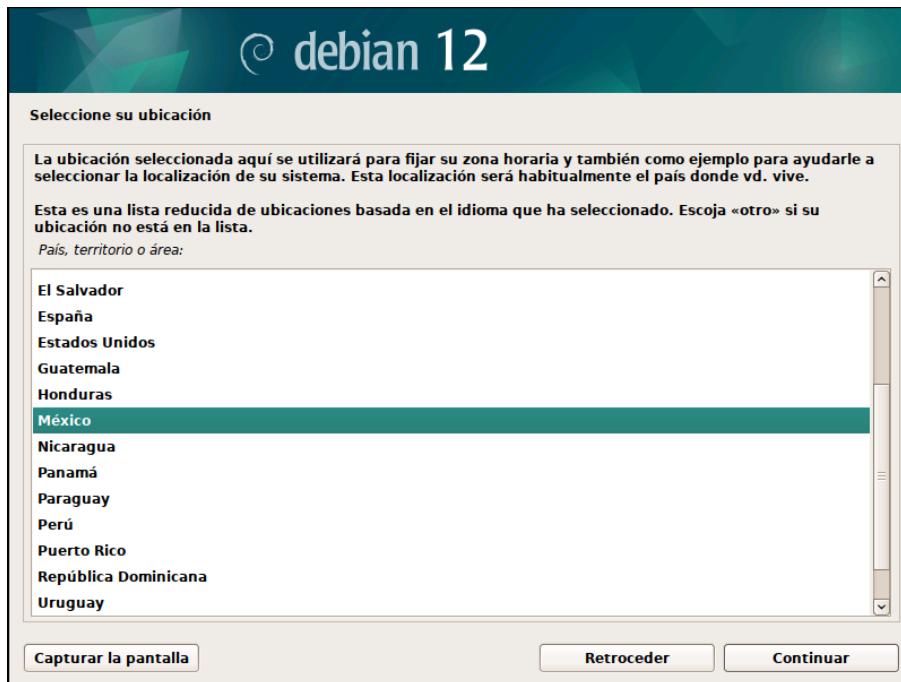


Figura 48. Seleccionamos nuestro país

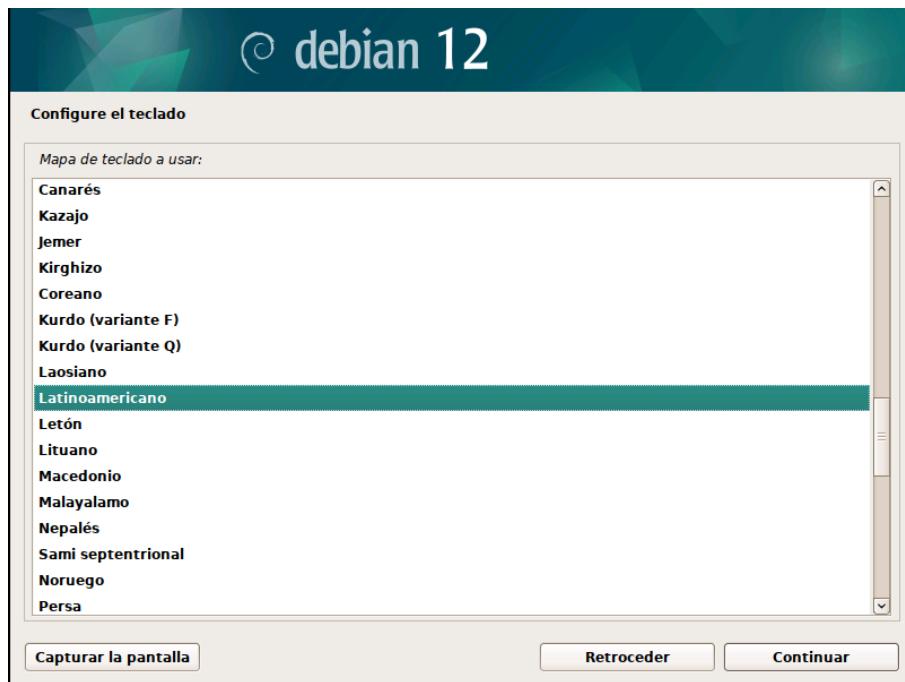


Figura 49. Seleccionamos nuestro teclado

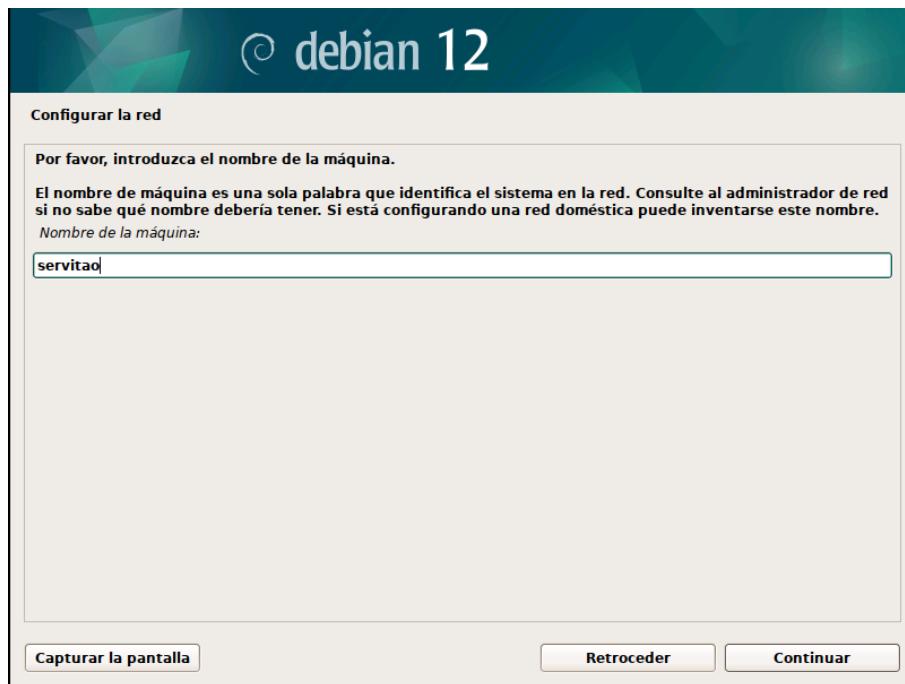


Figura 50. Le damos un nombre al servidor

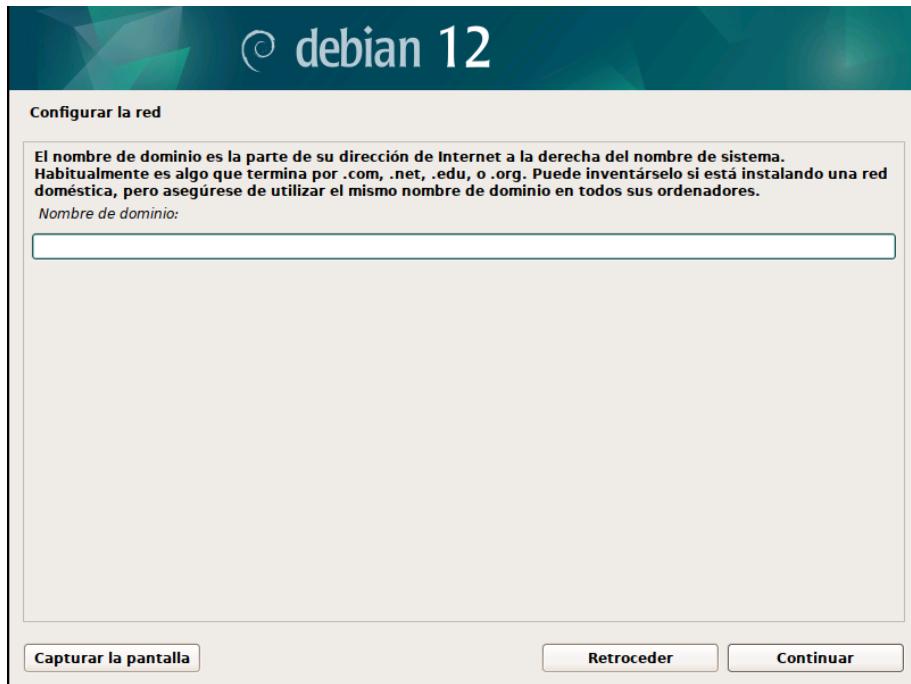


Figura 51. Si tenemos un dominio se lo damos, en mi caso no tengo uno así que lo dejo en blanco

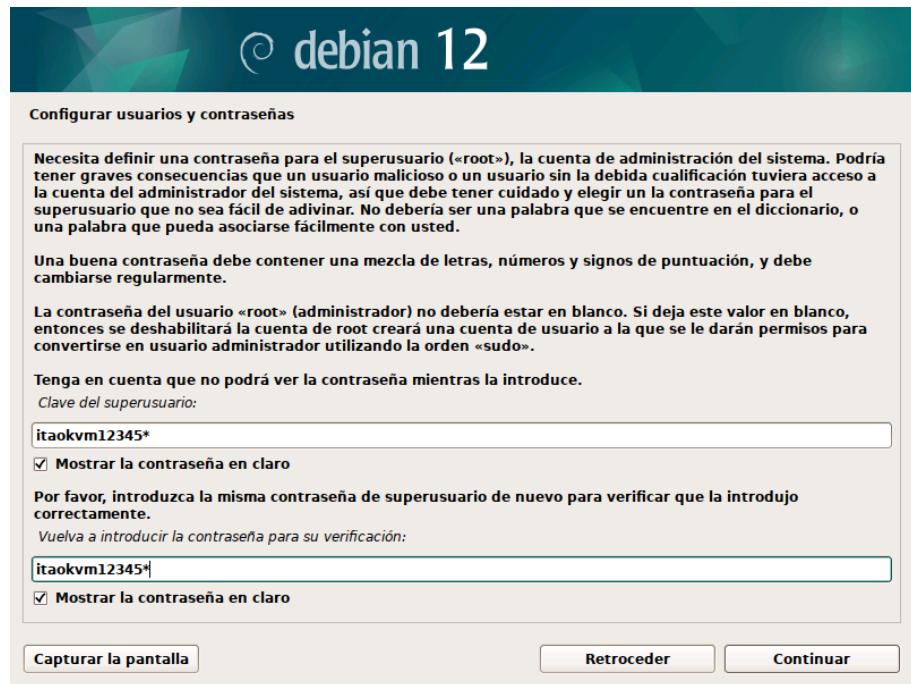


Figura 52. Le damos una contraseña en SU

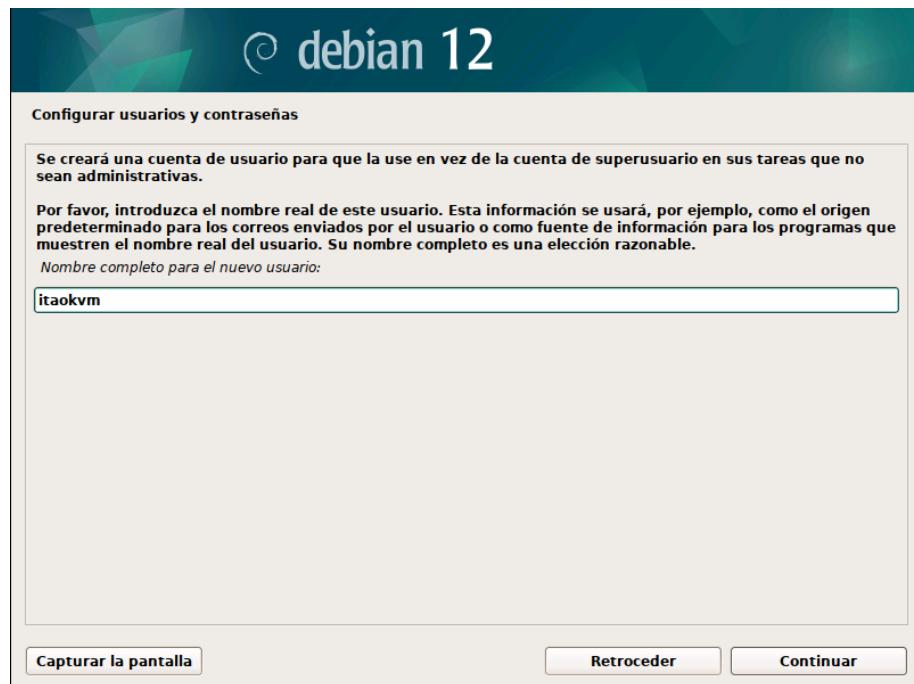


Figura 53. Le damos un nombre de usuario.

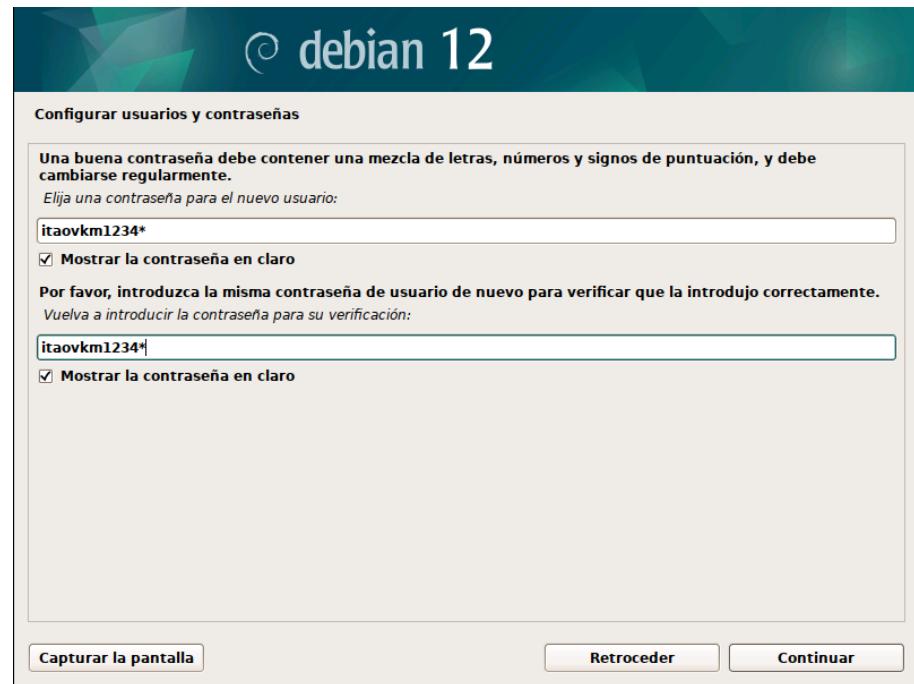


Figura 54. Le damos una contraseña al usuario.

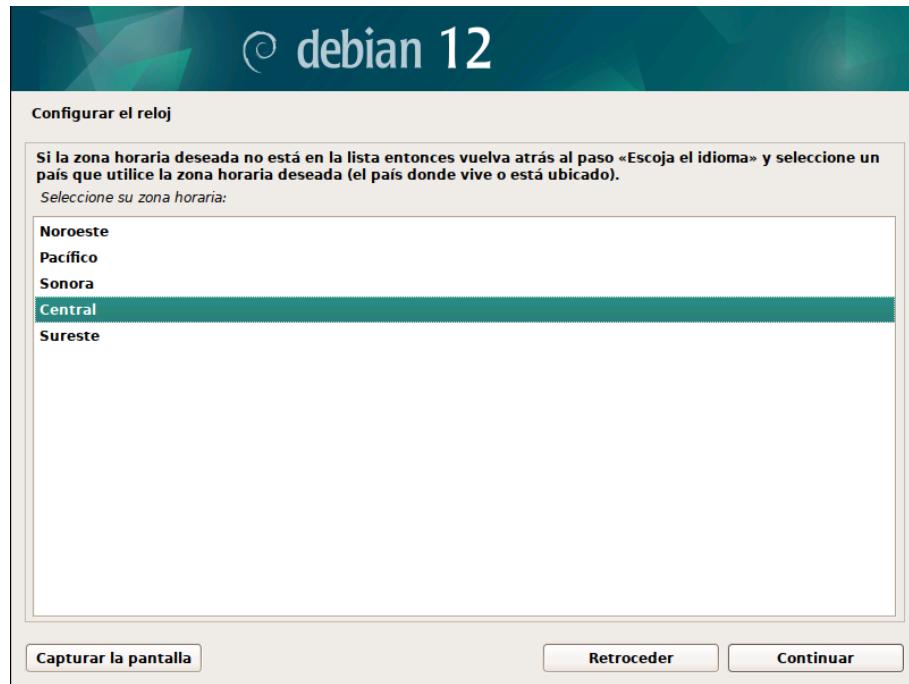


Figura 55. Escogemos la zona horaria, en mi caso es Central

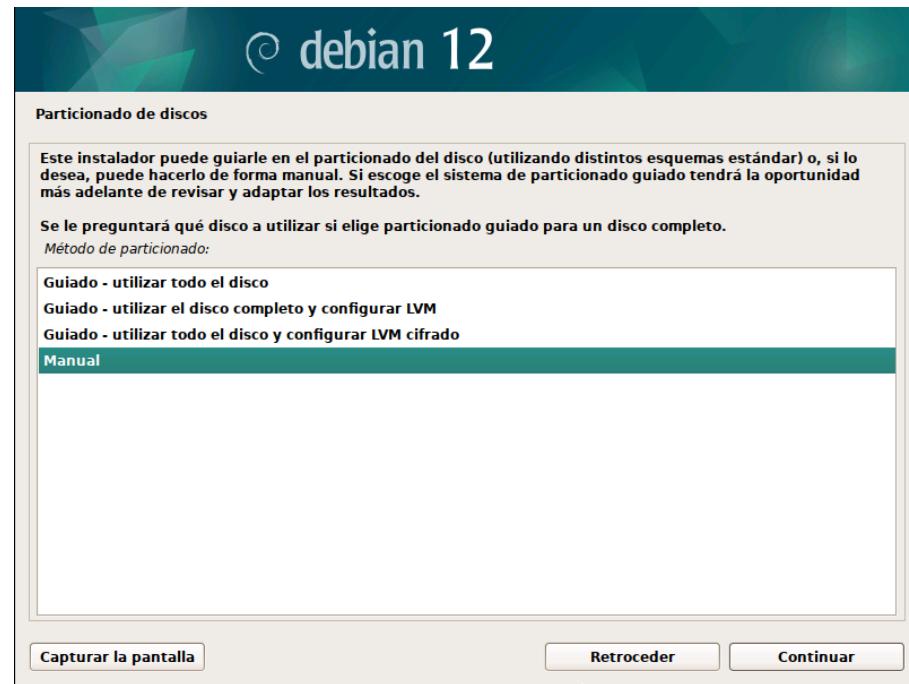


Figura 56. Seleccionamos manual para configurarlo a nuestra manera el disco

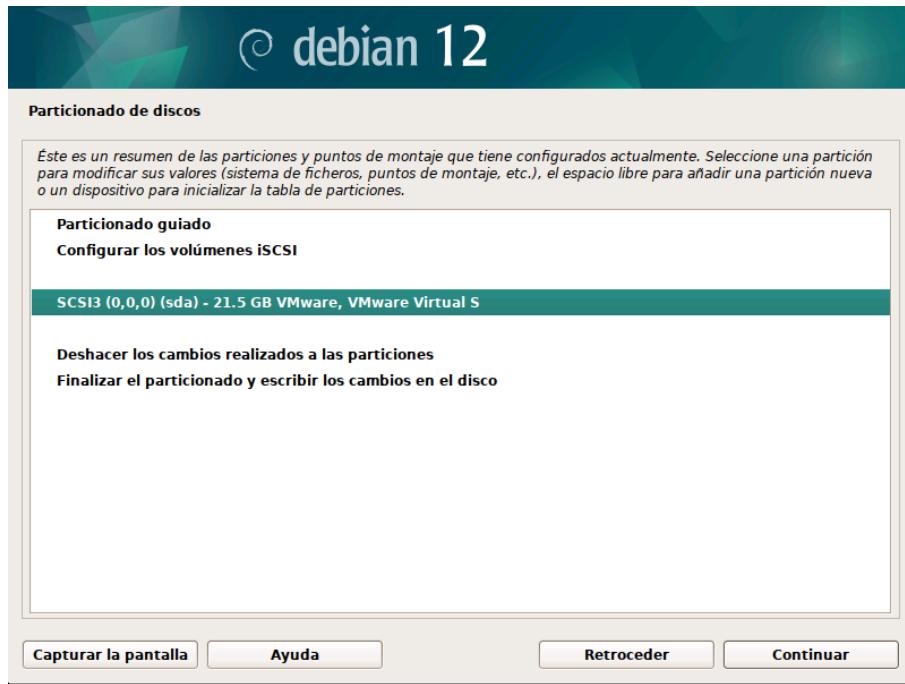


Figura 57. Seleccionamos nuestro disco duro

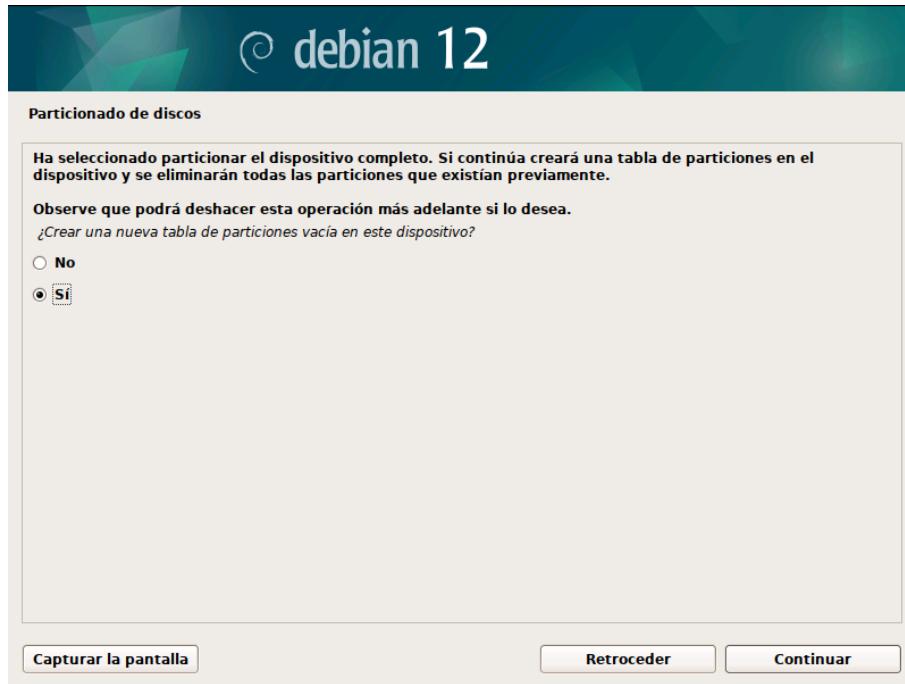


Figura 58. Le damos en si



Figura 59. Seleccionamos el espacio libre del disco



Figura 60. Se crea partición nueva



Figura 61. Seleccionamos el espacio para el SWAP

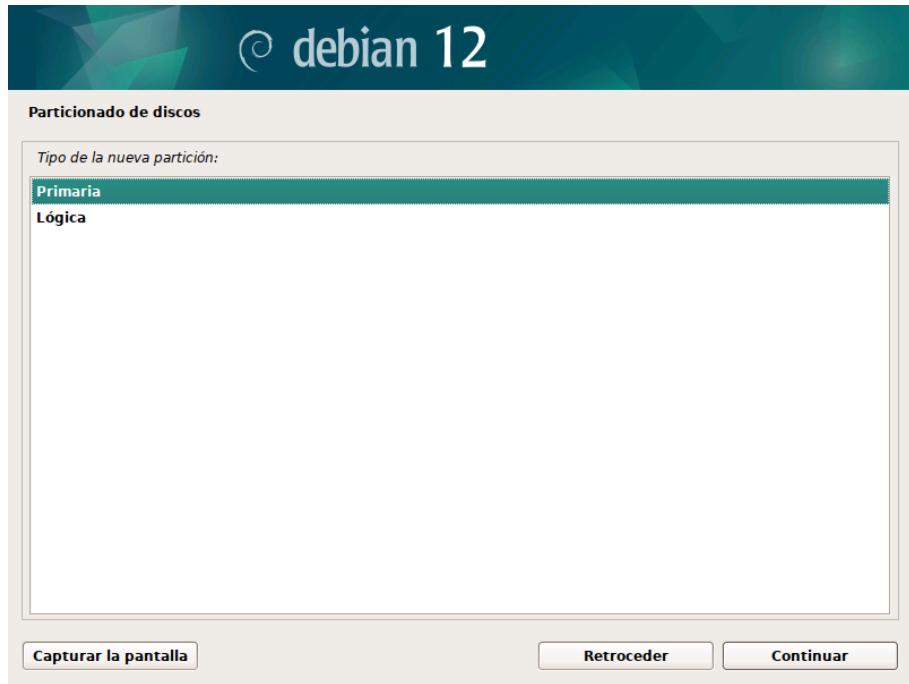


Figura 62. Seleccionamos primaria



Figura 63. cambiamos utilizar como:

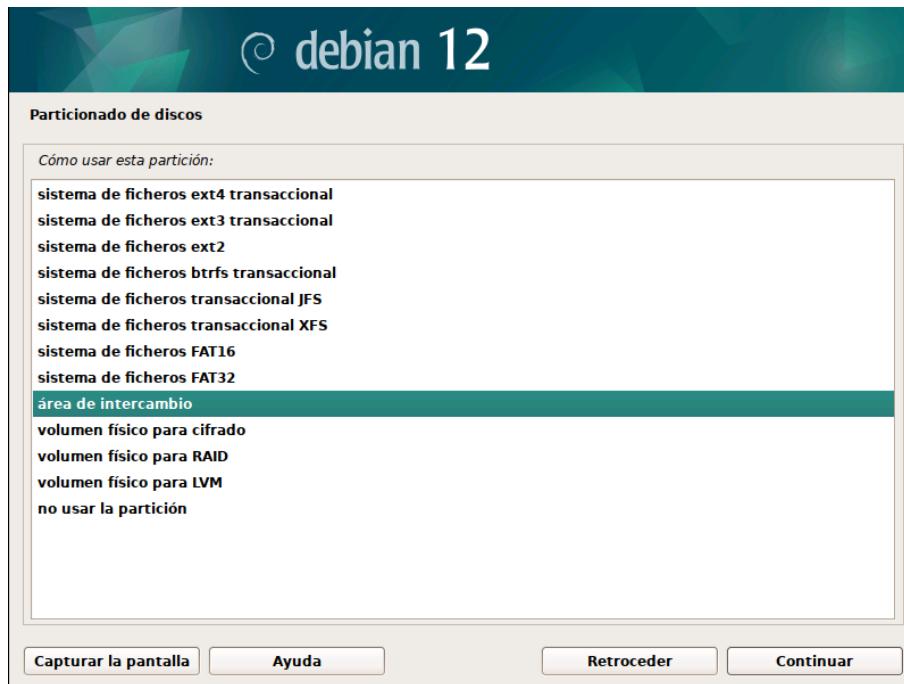


Figura 64. Seleccionamos el SWAP

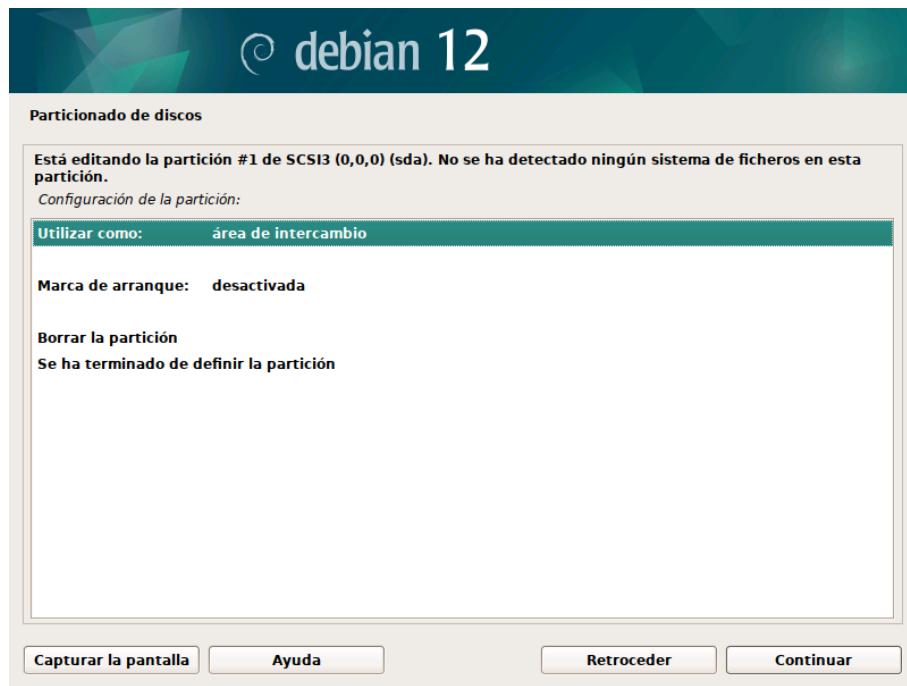


Figura 65. Aquí ya quedo seleccionado el SWAP



Figura 66. Hacemos lo mismo con el espacio que sobra, pero sobre raíz /



Figura 67. Con las particiones ya echas procedemos a finalizar el particionado

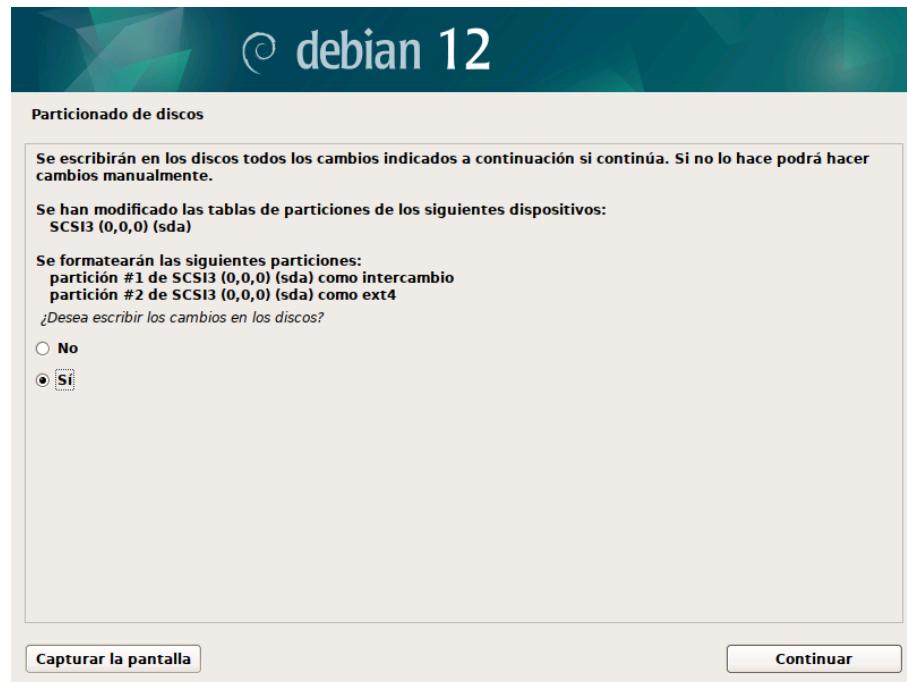


Figura 68. Le damos en si

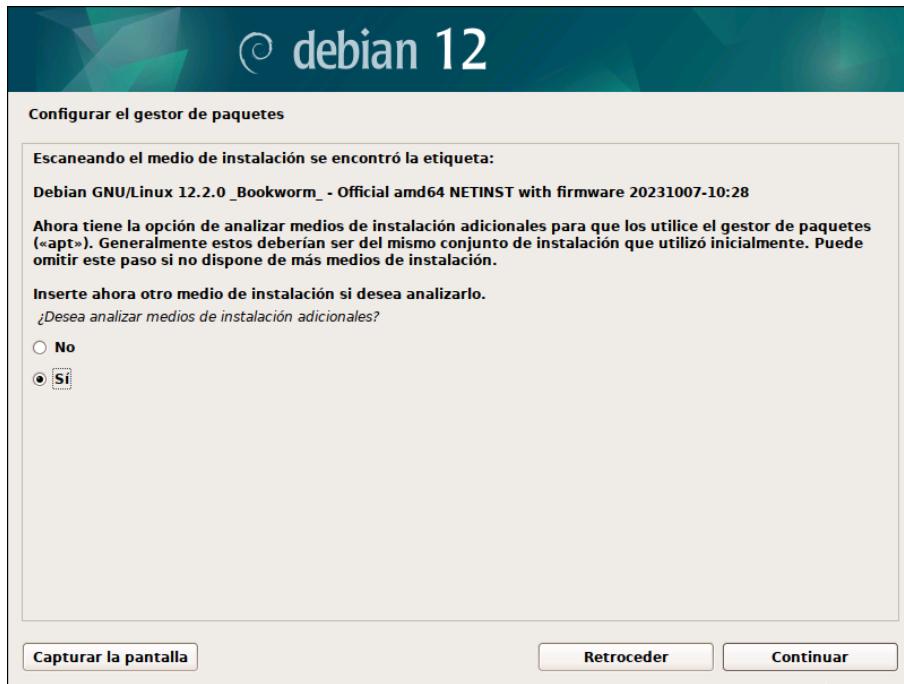


Figura 69. Le damos en si

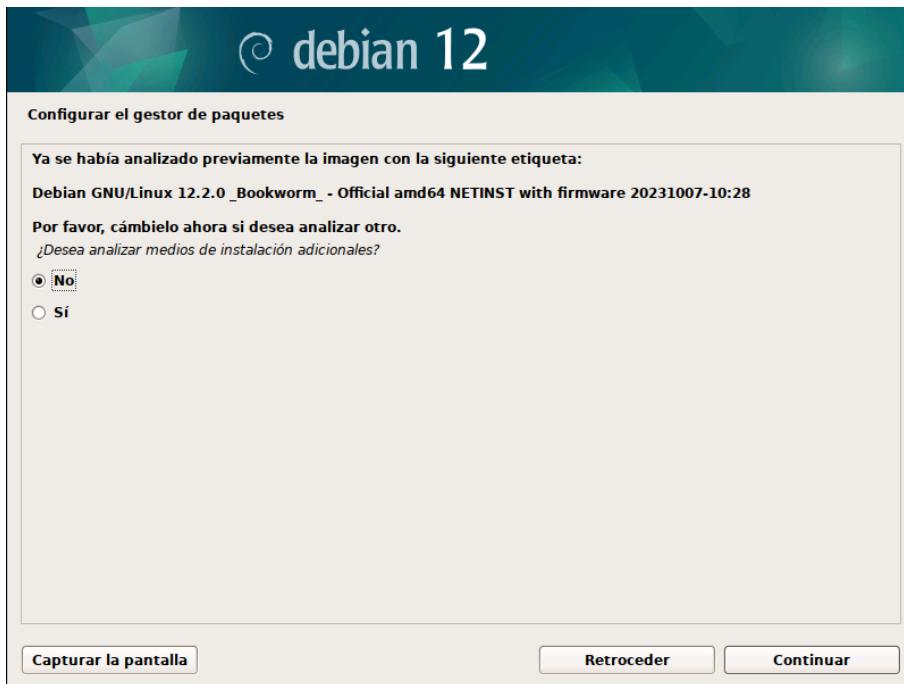


Figura 70. Una vez completado le damos en no

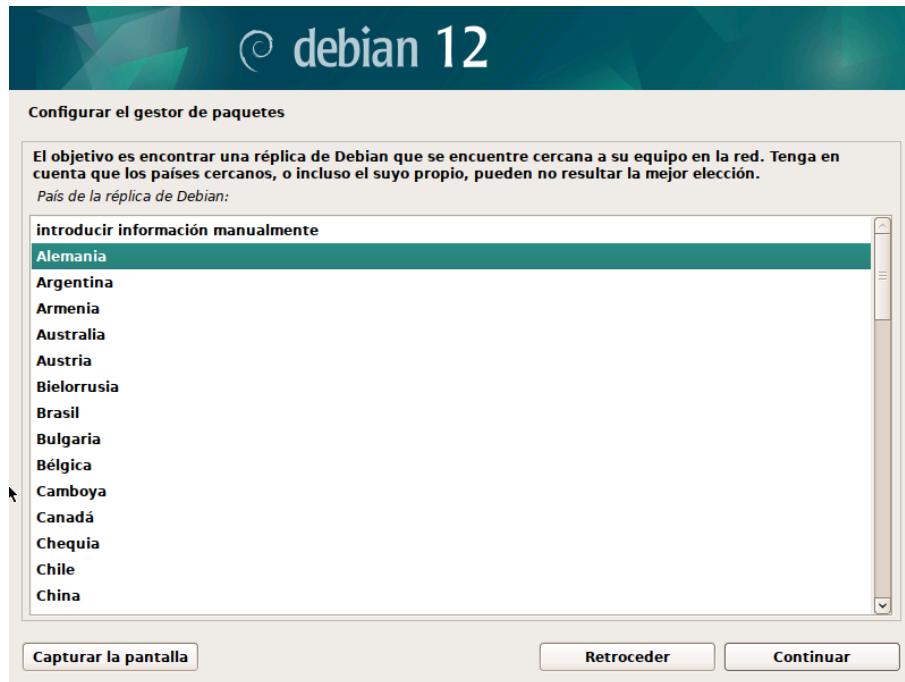


Figura 71. Seleccionamos Alemania por la zona horaria

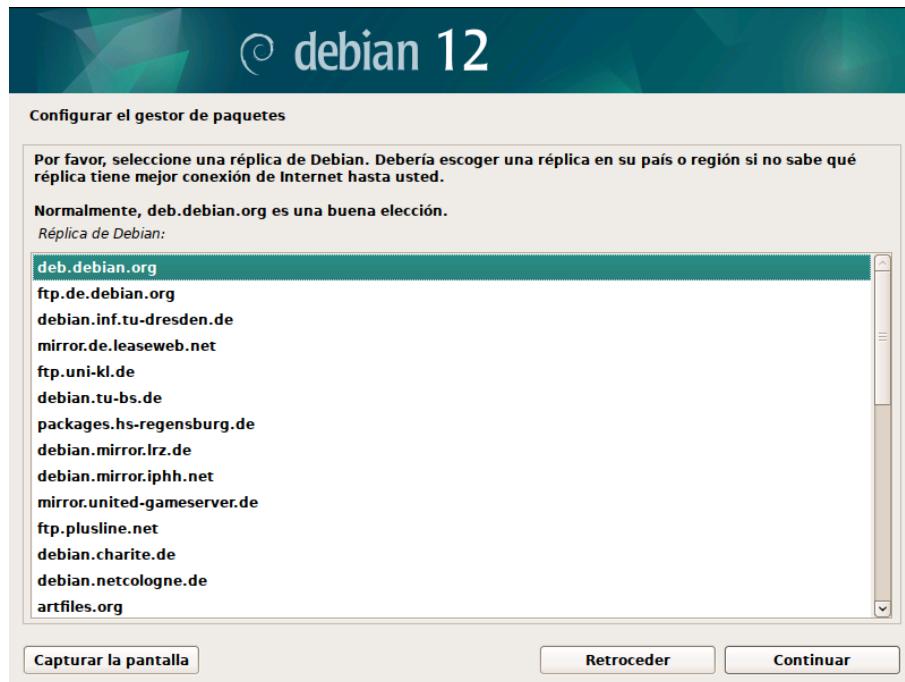


Figura 72. Dejamos la predeterminada

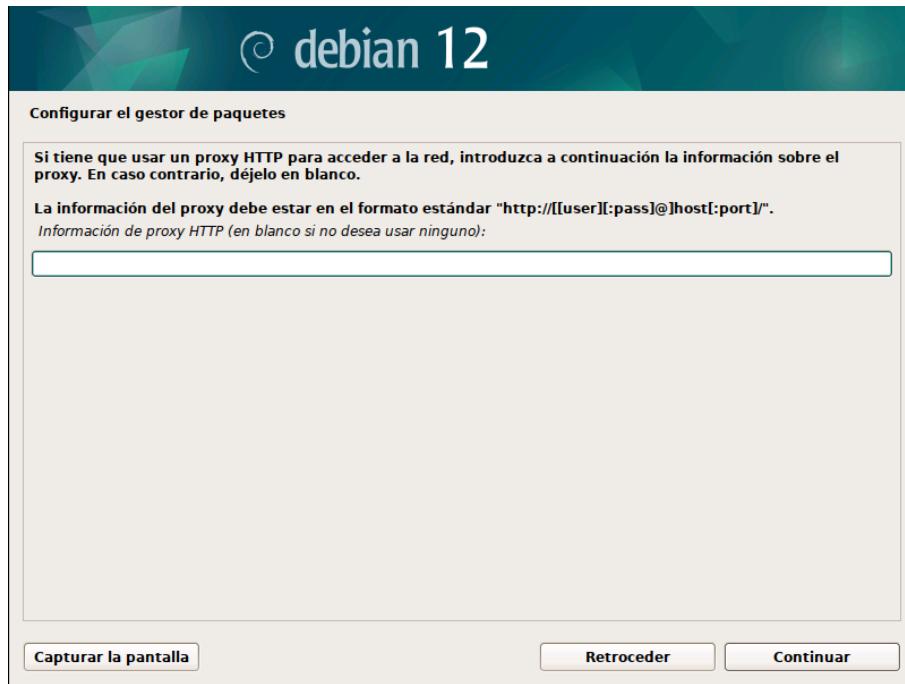


Figura 73. Si tienes un Proxy HTTP colócalo, pero si no déjalo en blanco y dale continuar

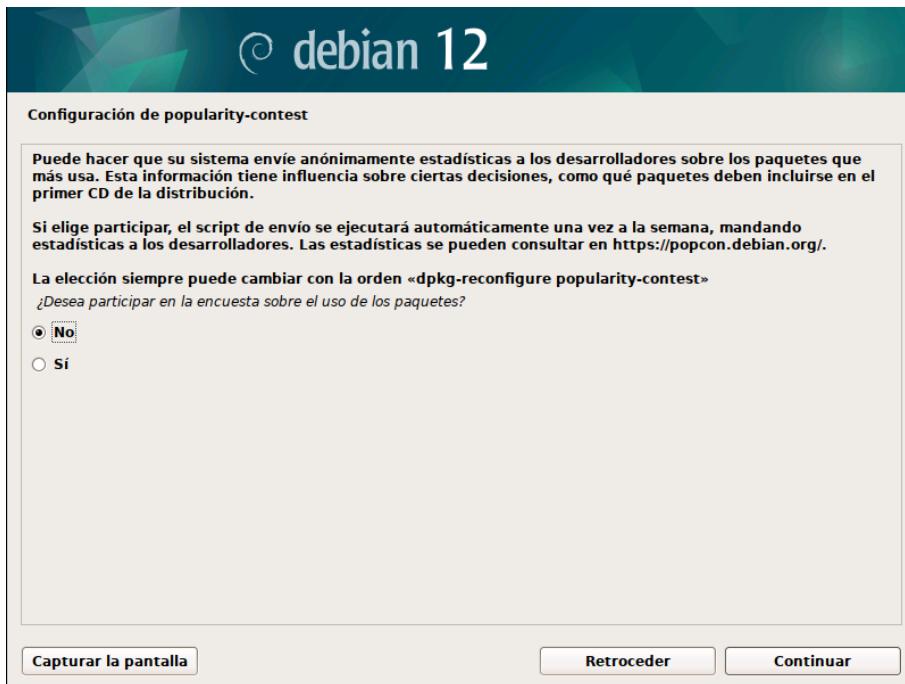


Figura 74. Esto depende de cada quien en mi caso le daré que no a la encuesta

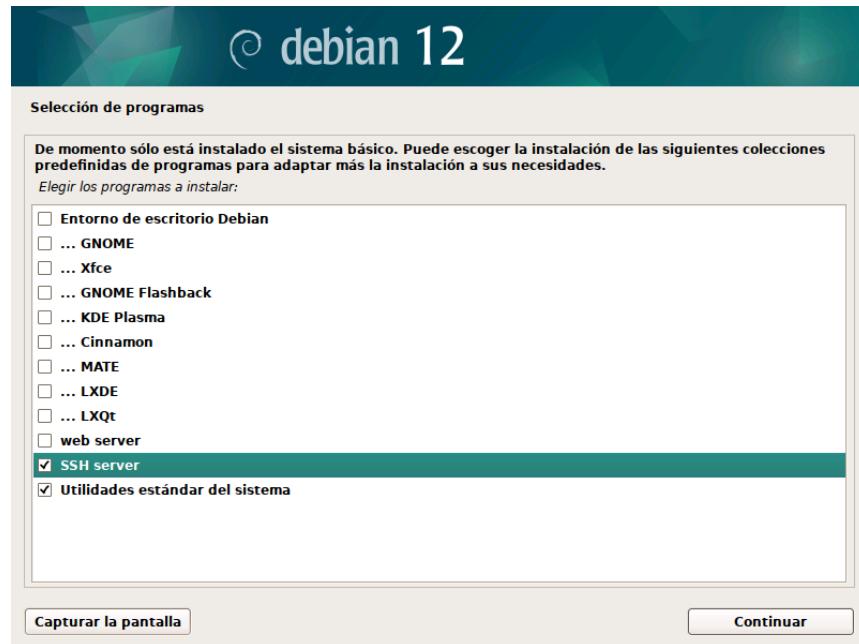


Figura 75. Lo configuramos de esta manera para que Debian corra en modo servidor y el ssh para usar PuTTY y controlar el servidor de mejor manera

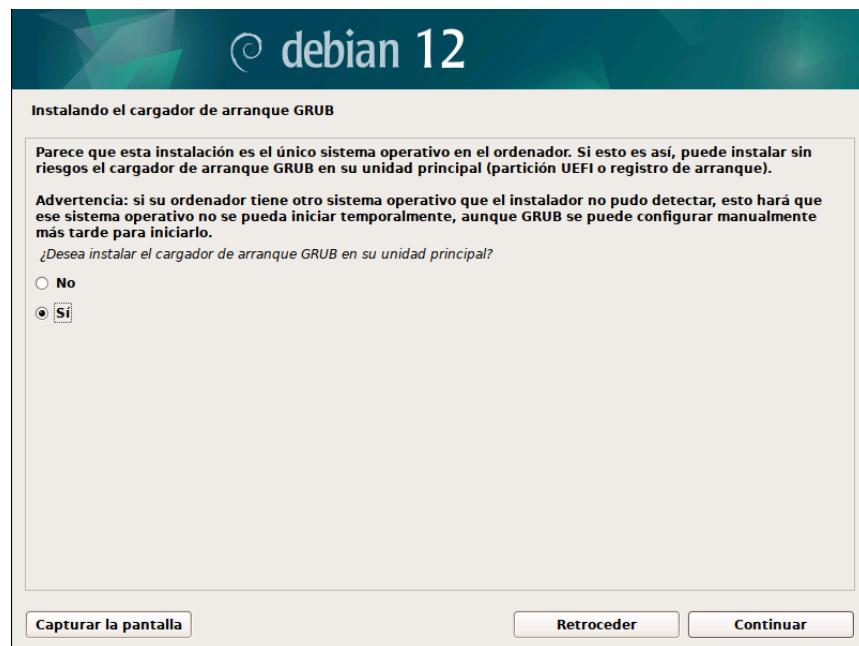


Figura 76. Le damos en si para instalar el GRUB de arranque

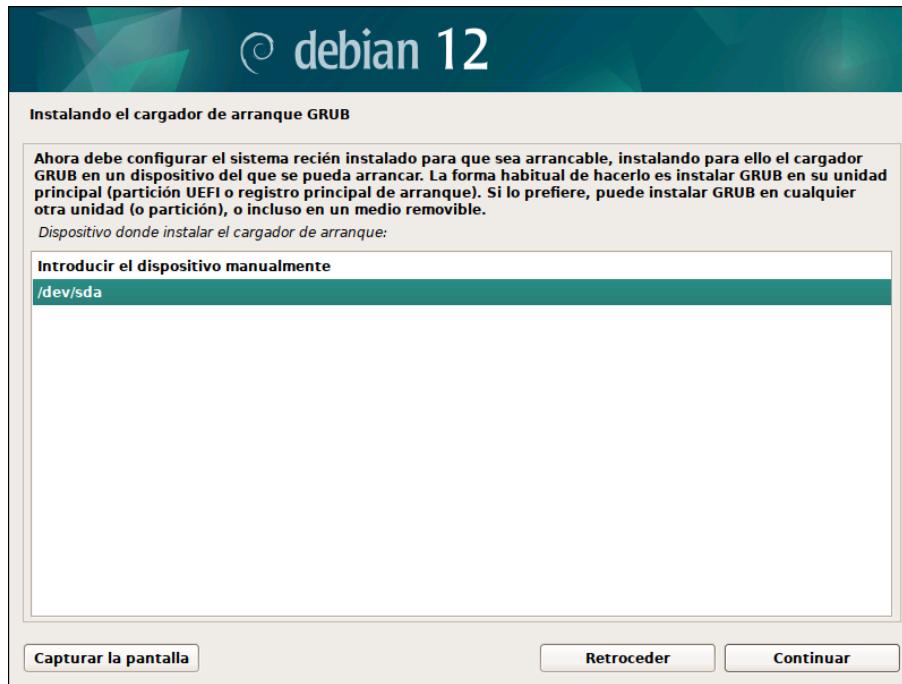


Figura 77. Seleccionamos el `/dev/sda` para que se instale el GRUB

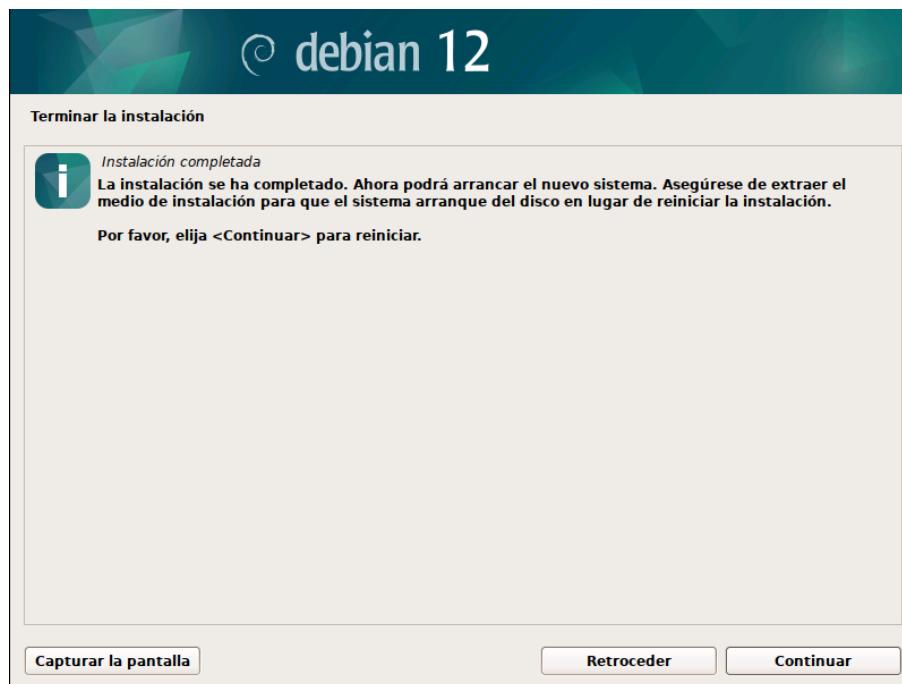


Figura 78. Le damos continuar para que se reinicie y se muestre el GRUB



Figura 79. Este es el GRUB de arranque de Debian 12 server

```
Debian GNU/Linux 12 servitao tty1

Hint: Num Lock on

servitao login: itaokvm
Password:
Linux servitao 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
itaokvm@servitao:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:3c:d8:f2 brd ff:ff:ff:ff:ff:ff
        altname enp2s1
        inet 192.168.127.133/24 brd 192.168.127.255 scope global dynamic ens33
            valid_lft 1443sec preferred_lft 1443sec
        inet6 fe80::20c:29ff:fe3c:d8f2/64 scope link
            valid_lft forever preferred_lft forever
itaokvm@servitao:~$
```

Figura 80. Solo iniciamos sesión y buscamos nuestra ip con #ip a y así conectarnos por PuTTY mediante ssh

Para establecer una conexión remota con el servidor, descargamos la aplicación PuTTY desde la URL: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Posteriormente, instalamos el paquete SSH en el servidor para permitir la conexión remota a través de PuTTY. Una vez instalado el paquete, abrimos la aplicación PuTTY y configuramos la conexión utilizando la dirección IP del servidor y un nombre de identificación específico.

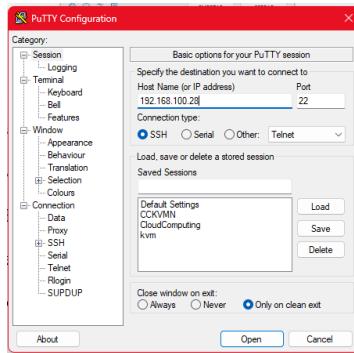


Figura 81. En la aplicación PuTTY colocamos la ip

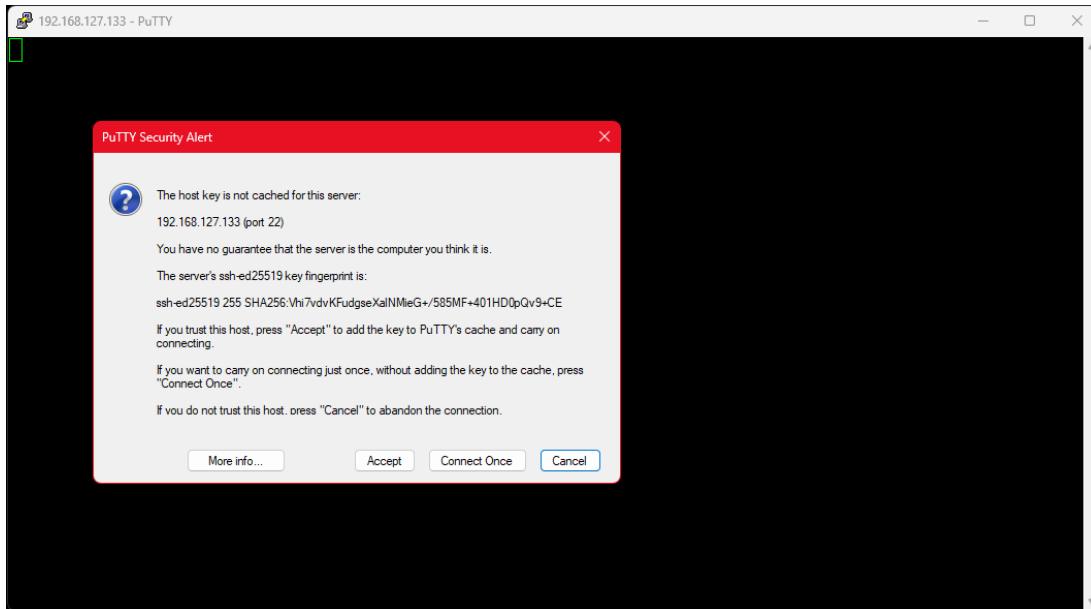
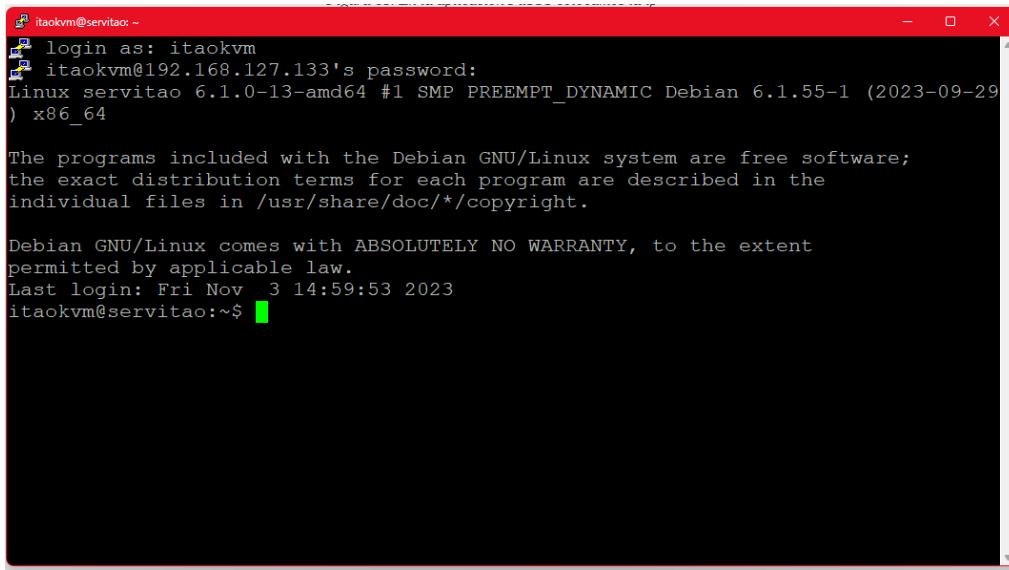


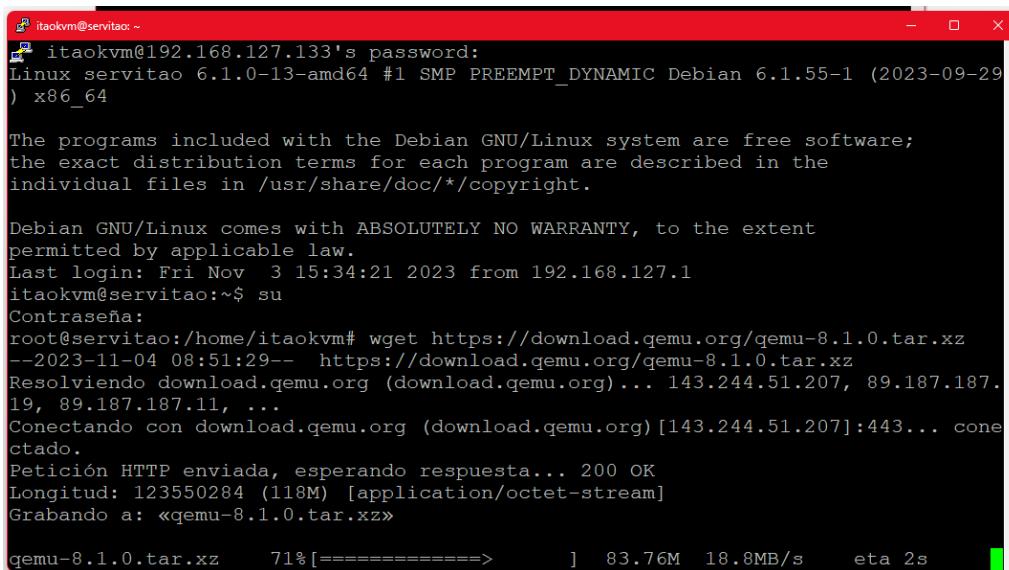
Figura 82. Le damos en aceptar



```
itaokvm@servitao:~  
login as: itaokvm  
itaokvm@192.168.127.133's password:  
Linux servitao 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29  
) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Nov 3 14:59:53 2023  
itaokvm@servitao:~$
```

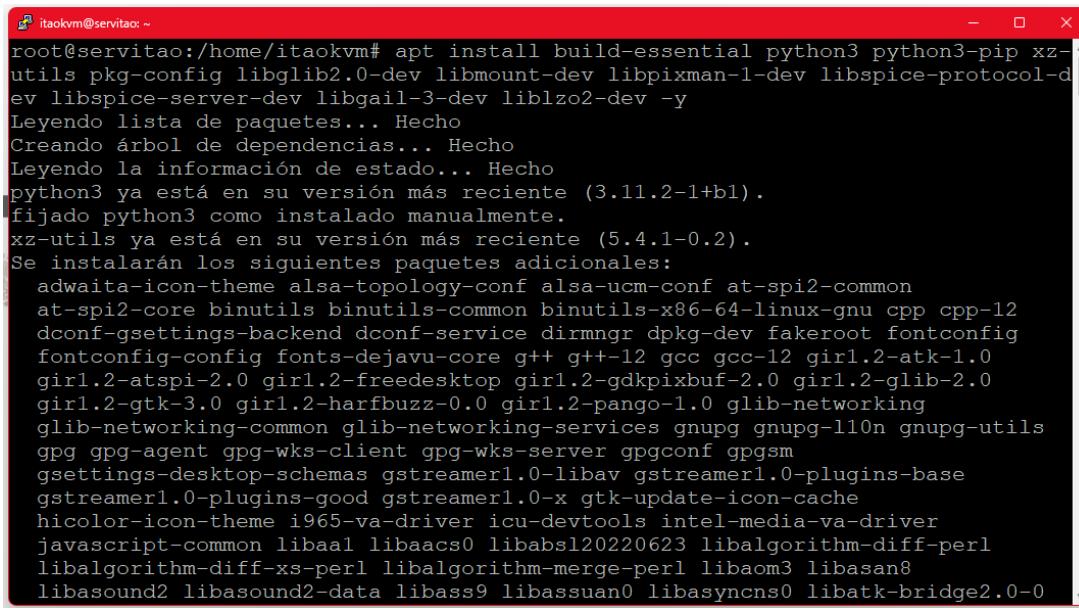
Figura 83. Aquí ya estamos dentro del servidor remotamente

Descargamos el repositorio de QEMU desde la URL: **#wget <https://download.qemu.org/qemu-8.1.0.tar.xz>** todo esto en modo SU. Luego, llevamos a cabo la instalación según los pasos indicados.



```
itaokvm@servitao:~  
itaokvm@192.168.127.133's password:  
Linux servitao 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29  
) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Nov 3 15:34:21 2023 from 192.168.127.1  
itaokvm@servitao:~$ su  
Contraseña:  
root@servitao:/home/itaokvm# wget https://download.qemu.org/qemu-8.1.0.tar.xz  
--2023-11-04 08:51:29-- https://download.qemu.org/qemu-8.1.0.tar.xz  
Resolviendo download.qemu.org (download.qemu.org)... 143.244.51.207, 89.187.187.  
19, 89.187.187.11, ...  
Conectando con download.qemu.org (download.qemu.org) [143.244.51.207]:443... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 123550284 (118M) [application/octet-stream]  
Grabando a: «qemu-8.1.0.tar.xz»  
  
qemu-8.1.0.tar.xz 71%[=====>] 83.76M 18.8MB/s eta 2s
```

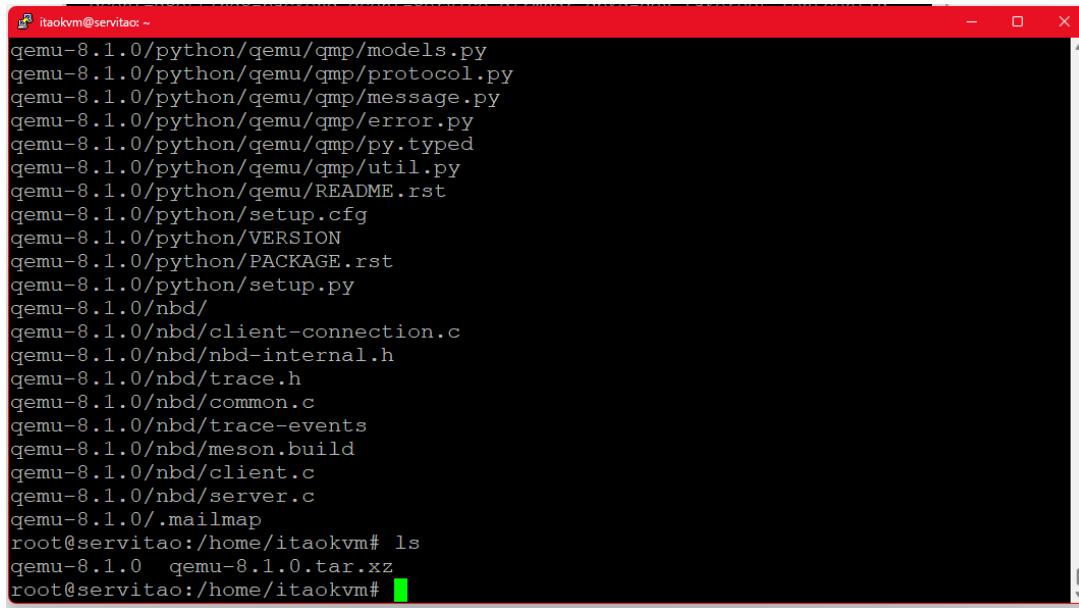
Figura 84. Descarga de QEMU



```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# apt install build-essential python3 python3-pip xz-utils pkg-config libglib2.0-dev libmount-dev libpixman-1-dev libspice-protocol-dev libspice-server-dev libgail-3-dev liblzo2-dev -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
python3 ya está en su versión más reciente (3.11.2-1+b1).
fijado python3 como instalado manualmente.
xz-utils ya está en su versión más reciente (5.4.1-0.2).
Se instalarán los siguientes paquetes adicionales:
adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-common
at-spi2-core binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-12
dconf-gsettings-backend dconf-service dirmngr dpkg-dev fakeroot fontconfig
fontconfig-config fonts-dejavu-core g++ g++-12 gcc gcc-12 gir1.2-atk-1.0
gir1.2-atspi-2.0 gir1.2-freedesktop gir1.2-gdkpixbuf-2.0 gir1.2-glib-2.0
gir1.2-gtk-3.0 gir1.2-harfbuzz-0.0 gir1.2-pango-1.0 glib-networking
glib-networking-common glib-networking-services gnupg gnupg-110n gnupg-utils
gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm
gsettings-desktop-schemas gstreamer1.0-libav gstreamer1.0-plugins-base
gstreamer1.0-plugins-good gstreamer1.0-x gtk-update-icon-cache
hicolor-icon-theme i965-va-driver icu-devtools intel-media-va-driver
javascript-common libaal libaacs0 libabsl20220623 libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libaom3 libasan8
libasound2 libasound2-data libass9 libassuan0 libasyncns0 libatk-bridge2.0-0
```

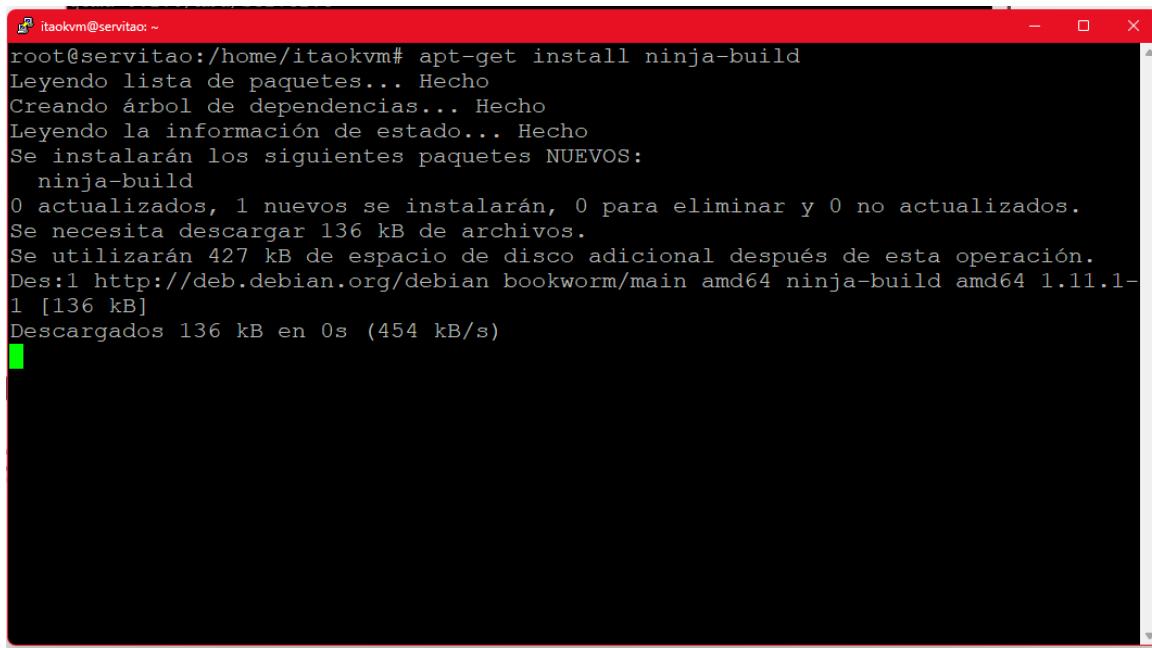
Figura 85. Se instala Python, SPICE entre otros paquetes necesarios para la virtualización. # *apt install build-essential python3*

python3-pip xz-utils pkg-config libglib2.0-dev libmount-dev libpixman-1-dev libspice-protocol-dev libspice-server-dev libgail-3-dev liblzo2-dev -y



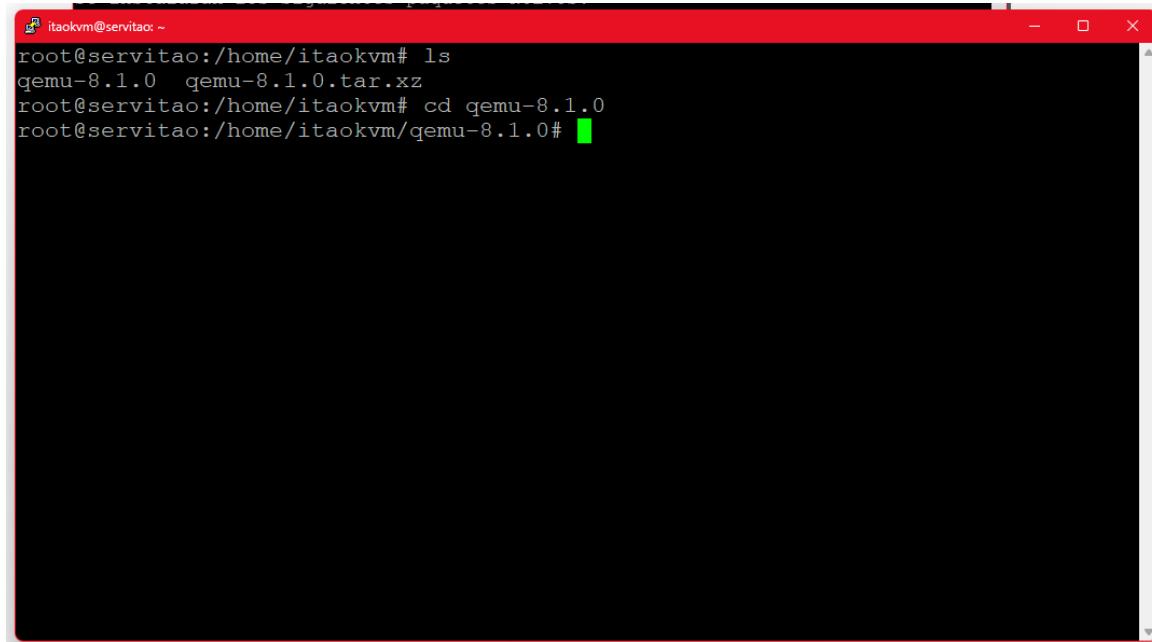
```
itaokvm@servitao: ~
qemu-8.1.0/python/qemu/qmp/models.py
qemu-8.1.0/python/qemu/qmp/protocol.py
qemu-8.1.0/python/qemu/qmp/message.py
qemu-8.1.0/python/qemu/qmp/error.py
qemu-8.1.0/python/qemu/qmp/py.typed
qemu-8.1.0/python/qemu/qmp/util.py
qemu-8.1.0/python/qemu/README.rst
qemu-8.1.0/python/setup.cfg
qemu-8.1.0/python/VERSION
qemu-8.1.0/python/PACKAGE.rst
qemu-8.1.0/python/setup.py
qemu-8.1.0/nbd/
qemu-8.1.0/nbd/client-connection.c
qemu-8.1.0/nbd/nbd-internal.h
qemu-8.1.0/nbd/trace.h
qemu-8.1.0/nbd/common.c
qemu-8.1.0/nbd/trace-events
qemu-8.1.0/nbd/meson.build
qemu-8.1.0/nbd/client.c
qemu-8.1.0/nbd/server.c
qemu-8.1.0/.mailmap
root@servitao:/home/itaokvm# ls
qemu-8.1.0 qemu-8.1.0.tar.xz
root@servitao:/home/itaokvm#
```

Figura 86. Descomprimimos el archivo con # *tar Jvxf qemu-8.1.0.tar.xz*



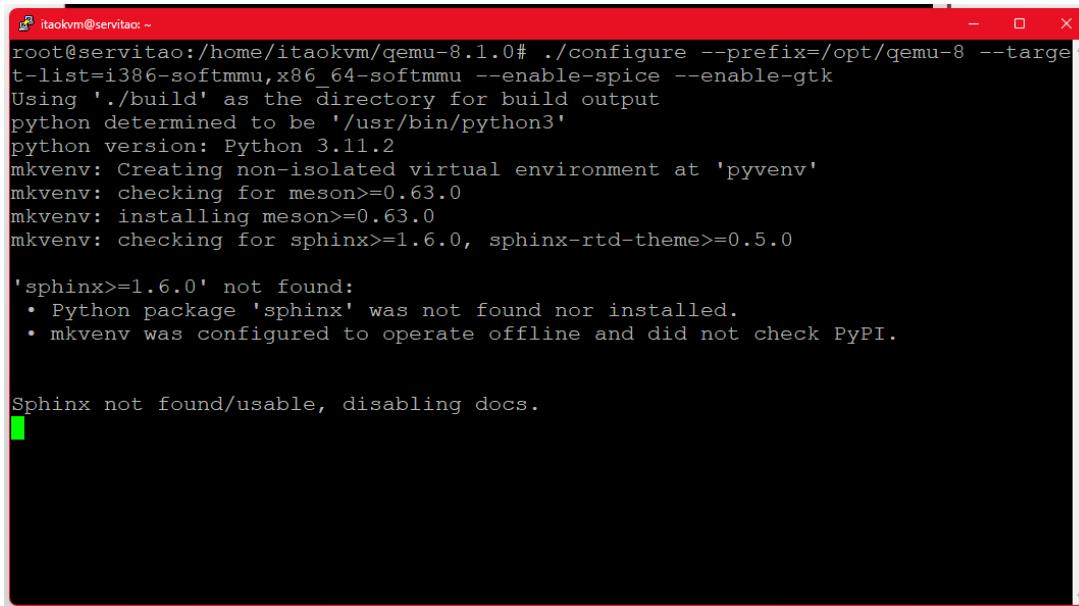
```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# apt-get install ninja-build
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  ninja-build
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 136 kB de archivos.
Se utilizarán 427 kB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 ninja-build amd64 1.11.1-1 [136 kB]
Descargados 136 kB en 0s (454 kB/s)
```

Figura 87. Instalamos el paquete *ninja* para construir el programa de QEMU con # *apt-get install ninja-build*



```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# ls
qemu-8.1.0  qemu-8.1.0.tar.xz
root@servitao:/home/itaokvm# cd qemu-8.1.0
root@servitao:/home/itaokvm/qemu-8.1.0#
```

Figura 88. Nos movemos a la carpeta descomprimida con el siguiente comando # *cd qemu-8.1.0*

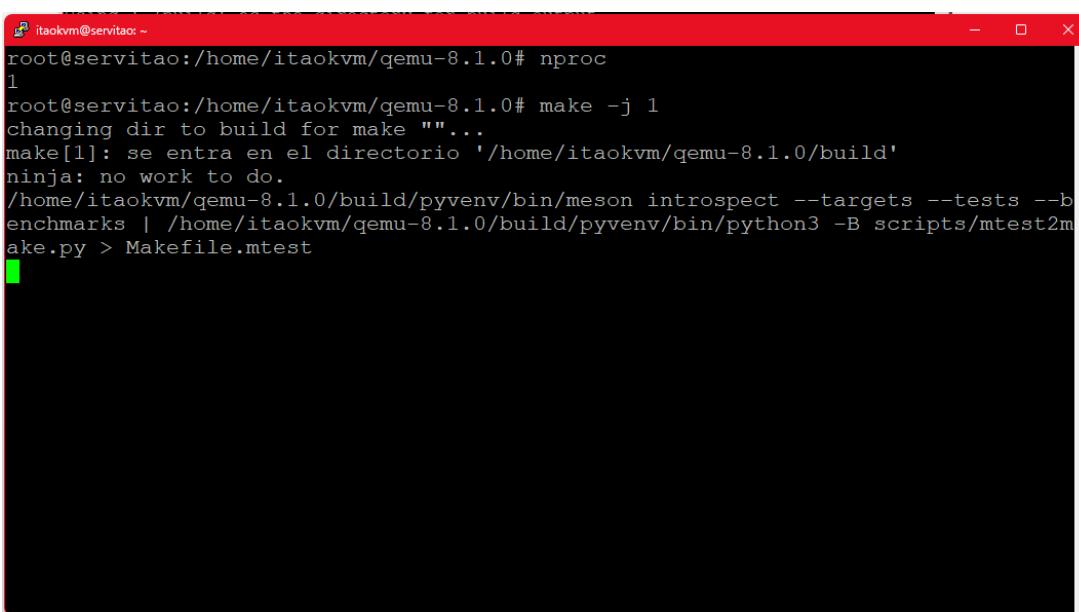


```
itaokvm@servitao: ~
root@servitao:/home/itaokvm/qemu-8.1.0# ./configure --prefix=/opt/qemu-8 --target-list=i386-softmmu,x86_64-softmmu --enable-spice --enable-gtk
Using './build' as the directory for build output
python determined to be '/usr/bin/python3'
python version: Python 3.11.2
mkvenv: Creating non-isolated virtual environment at 'pyvenv'
mkvenv: checking for meson>=0.63.0
mkvenv: installing meson>=0.63.0
mkvenv: checking for sphinx>=1.6.0, sphinx-rtd-theme>=0.5.0

'sphinx>=1.6.0' not found:
• Python package 'sphinx' was not found nor installed.
• mkvenv was configured to operate offline and did not check PyPI.

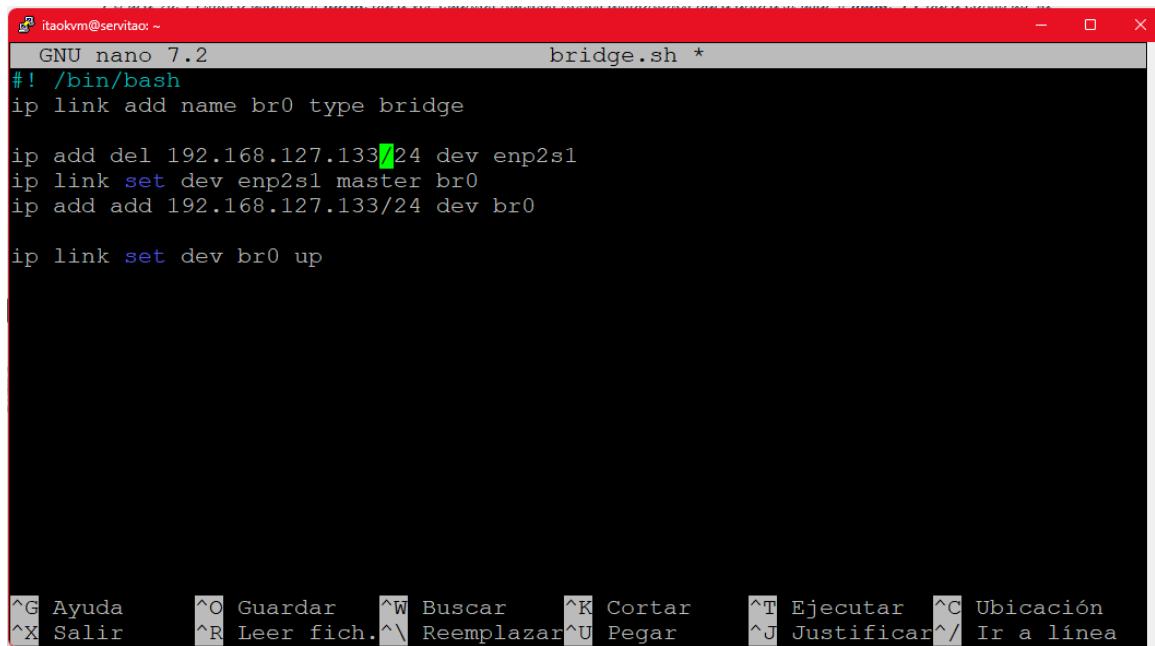
Sphinx not found/usable, disabling docs.
```

Figura 89. primero configuramos el compilador para la construccion del software con el siguiente comando `# ./configure—prefix=/opt/qemu-8 --target-list=i386-softmmu,x86_64-softmmu—enable-spice—enable-gtk`



```
itaokvm@servitao: ~
root@servitao:/home/itaokvm/qemu-8.1.0# nproc
1
root@servitao:/home/itaokvm/qemu-8.1.0# make -j 1
changing dir to build for make """
make[1]: se entra en el directorio '/home/itaokvm/qemu-8.1.0/build'
ninja: no work to do.
/home/itaokvm/qemu-8.1.0/build/pyvenv/bin/meson introspect --targets --tests --benchmarks | /home/itaokvm/qemu-8.1.0/build/pyvenv/bin/python3 -B scripts/mtest2make.py > Makefile.mtest
```

Figura 90. Primero usamos `# nproc` para ver cuántos núcleos tienes disponibles para ahora si usar `# make -j 1` para construir la aplicación.



itaokvm@servitao: ~

GNU nano 7.2 bridge.sh *

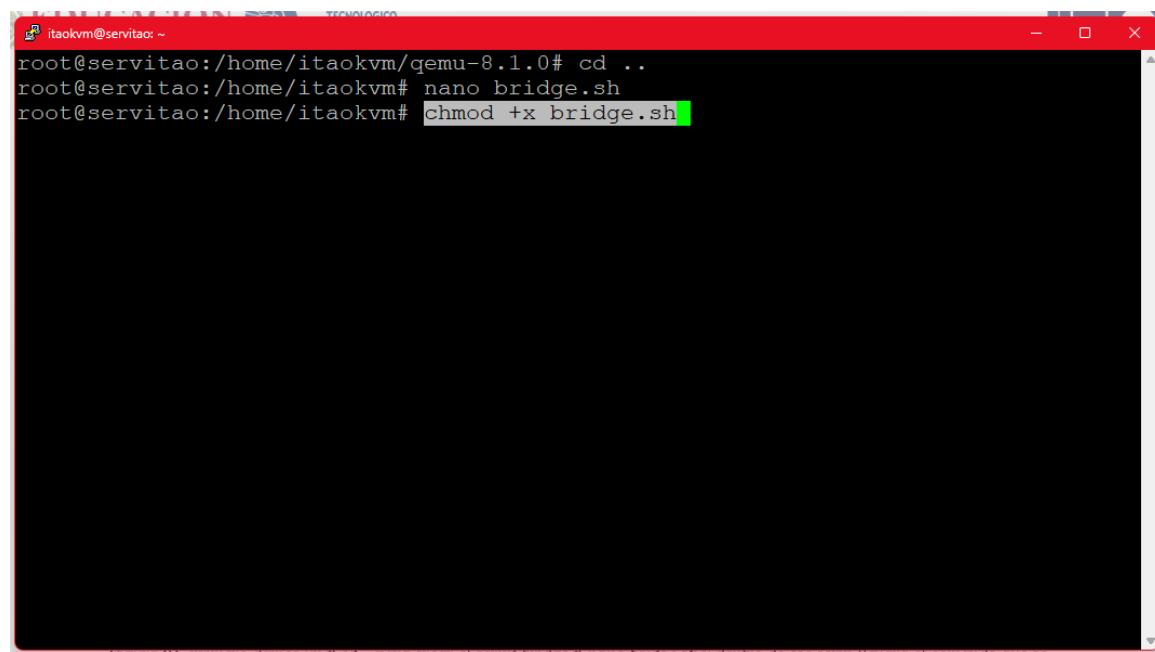
```
#!/bin/bash
ip link add name br0 type bridge

ip add del 192.168.127.133/24 dev enp2s1
ip link set dev enp2s1 master br0
ip add add 192.168.127.133/24 dev br0

ip link set dev br0 up
```

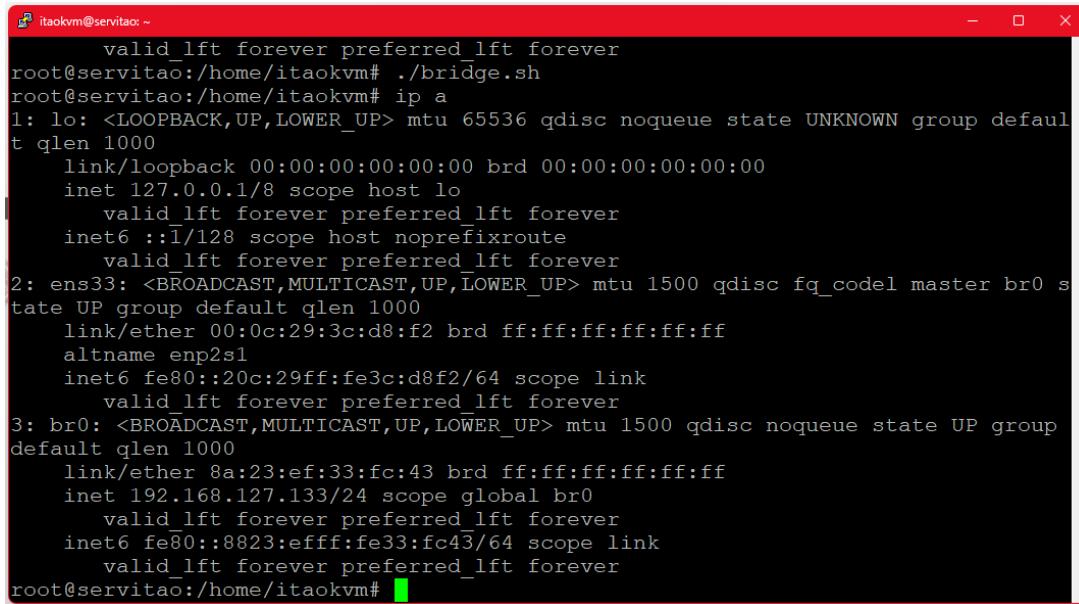
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación
^X Salir ^R Leer fich.^` Reemplazar^U Pegar ^J Justificar^/ Ir a línea

Figura 91. primero damos un # cd .. para crear el script bridge # nano bridge.sh y dentro de ese scrip llevara el comando que se ve en la figura mostrada, hay que cambiar la ip a la de ustedes, guardamos el script con ctrl + o



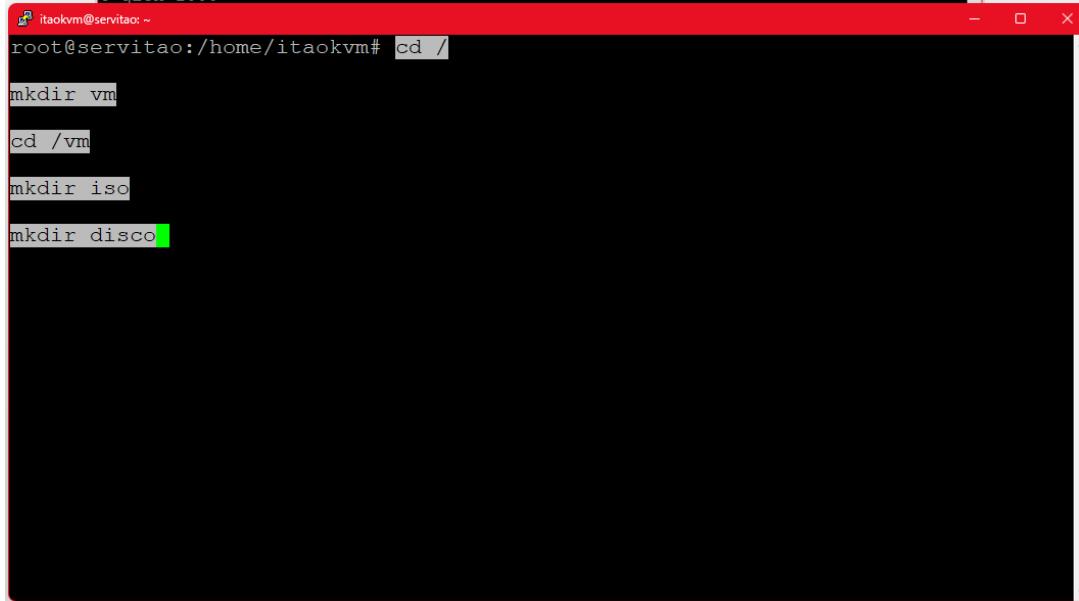
```
itaokvm@servitao: ~
root@servitao:/home/itaokvm/qemu-8.1.0# cd ..
root@servitao:/home/itaokvm# nano bridge.sh
root@servitao:/home/itaokvm# chmod +x bridge.sh
```

Figura 92. le aplicamos el comando de ejecución con # chmod +x bridge.sh



```
itaokvm@servitao: ~
valid_lft forever preferred_lft forever
root@servitao:/home/itaokvm# ./bridge.sh
root@servitao:/home/itaokvm# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master br0 state UP group default qlen 1000
    link/ether 00:0c:29:3c:d8:f2 brd ff:ff:ff:ff:ff:ff
    altnet enp2s1
    inet6 fe80::20c:29ff:fe3c:d8f2/64 scope link
        valid_lft forever preferred_lft forever
3: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 8a:23:ef:33:fc:43 brd ff:ff:ff:ff:ff:ff
    inet 192.168.127.133/24 scope global br0
        valid_lft forever preferred_lft forever
    inet6 fe80::8823:effff:fe33:fc43/64 scope link
        valid_lft forever preferred_lft forever
root@servitao:/home/itaokvm#
```

Figura 93. vamos a ejecutar el script # ./bridge.sh y después # ip a para ver si se realizaron los cambios.

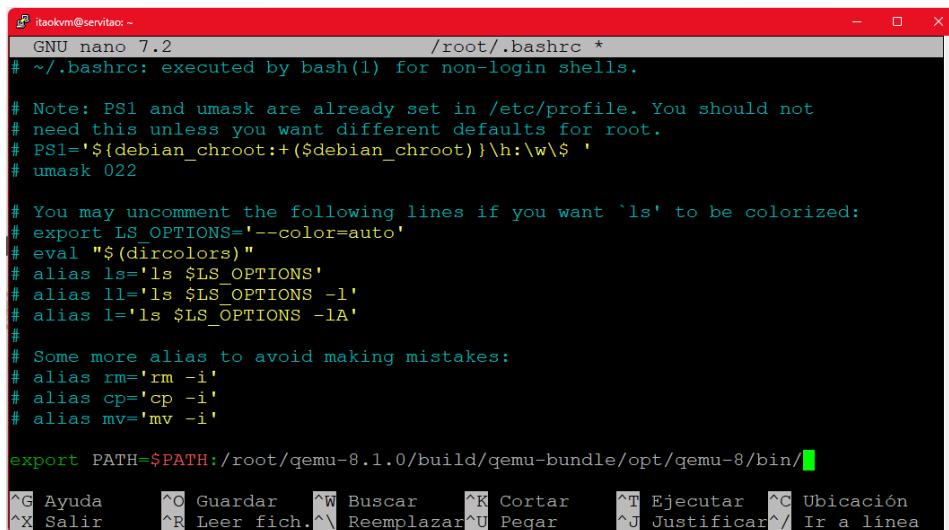


```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# cd /
mkdir vm
cd /vm
mkdir iso
mkdir disco
```

Figura 94. nos movemos a la carpeta raíz # cd / despues creamos un directorio con # mkdir vm nos movemos a la carpeta creada con # cd /vm y volvemos a crear 2 directorios más como se ven en la figura.

```
root@servitao:/vm# find / -name qemu-img 2>/dev/null
/home/itaokvm/qemu-8.1.0/build/qemu-img
/home/itaokvm/qemu-8.1.0/build/qemu-bundle/opt/qemu-8/bin/qemu-img
root@servitao:/vm#
```

Figura 95. buscamos la ruta para meterla en la variable de entorno y se pueda ejecutar, usamos el siguiente comando `#find / -name qemu-img 2>/dev/null`



```
GNU nano 7.2          /root/.bashrc *
# ~./bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want 'ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -la'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'

export PATH=$PATH:/root/qemu-8.1.0/build/qemu-bundle/opt/qemu-8/bin/
```

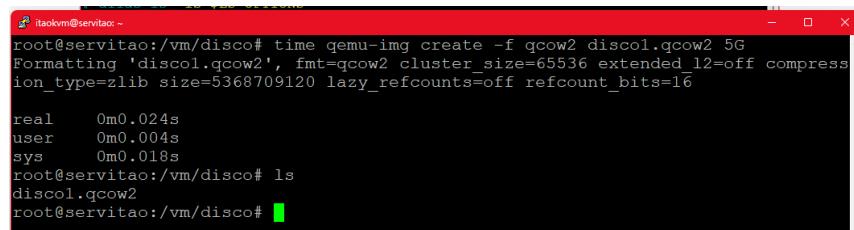
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación
^X Salir ^R Leer fich. ^V Reemplazar ^U Pegar ^J Justificar ^/ Ir a linea

Figura 96. ejecutamos `# nano ~./bashrc` para abrir el PATH y asignar nuestra variable de entorno `export`

`PATH=$PATH:/root/qemu-8.1.0/build/qemu-bundle/opt/qemu-8/bin/` guardamos con `ctrl + o` y cerramos con `ctrl + x`.

```
root@servitao:/vm# nano ~./bashrc
root@servitao:/vm# source ~./bashrc
```

Figura 97. Guardamos los cambios con `# source ~./bashrc`



```
root@servitao:/vm/disco# time qemu-img create -f qcow2 disco1.qcow2 5G
Formatting 'disco1.qcow2', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=5368709120 lazy_refcounts=off refcount_bits=16
real    0m0.024s
user    0m0.004s
sys     0m0.018s
root@servitao:/vm/disco# ls
disco1.qcow2
root@servitao:/vm/disco#
```

Figura 98. creamos un disco virtual para almacenar el sistema operativo, pero primero nos movemos a la carpeta `# cd disco/`

despues ejecutamos el comando `# qemu-img create -f qcow2 disco1.qcow2 20G`

Instituto Tecnológico de Álvaro Obregón

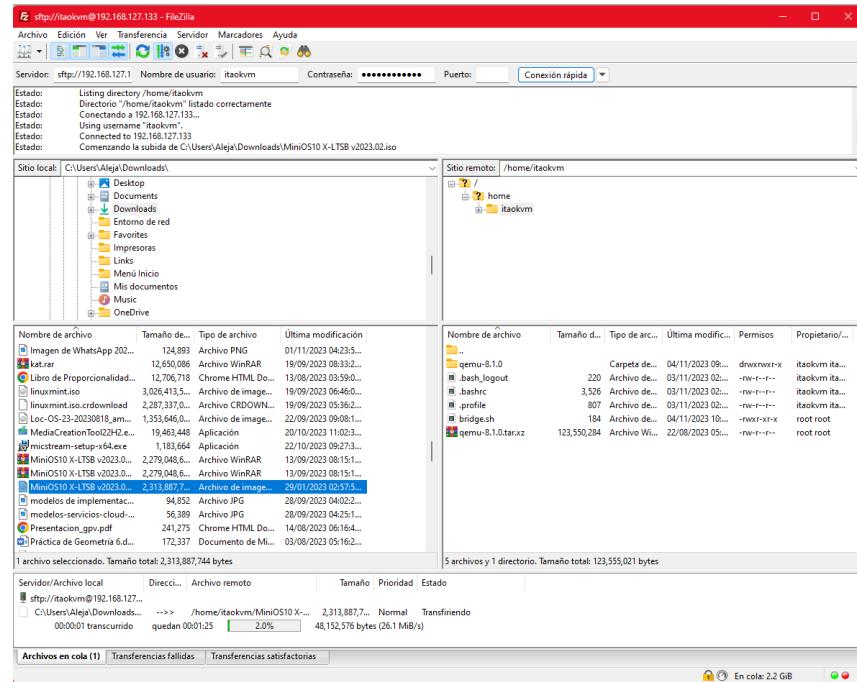
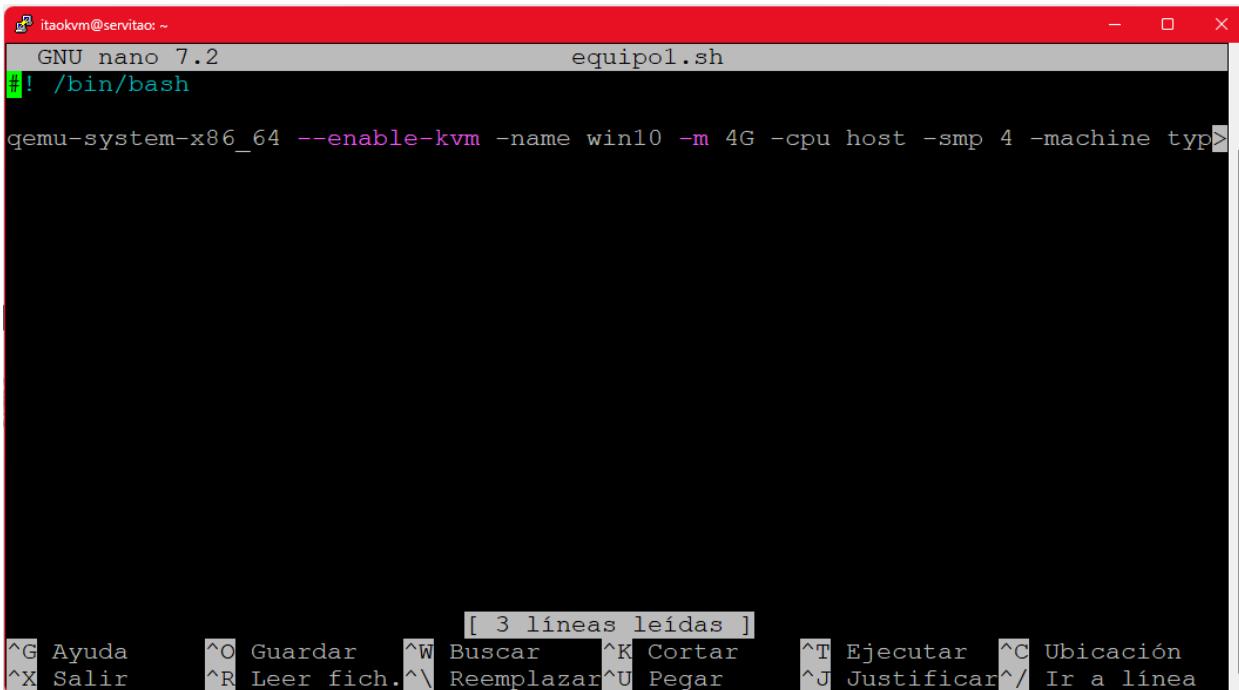


Figura 99. Movemos el ISO mediante FileZilla

```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# ls
bridge.sh 'MiniOS10 X-LTSB v2023.02.iso' qemu-8.1.0 qemu-8.1.0.tar.xz
root@servitao:/home/itaokvm# mv MiniOS10\ X-LTSB\ v2023.02.iso /vm/iso/win10
root@servitao:/home/itaokvm# ls
bridge.sh qemu-8.1.0 qemu-8.1.0.tar.xz
root@servitao:/home/itaokvm# cd /vm/iso/
root@servitao:/vm/iso# ls
win10
root@servitao:/vm/iso# mv win10 win10.iso
root@servitao:/vm/iso# ls
win10.iso
root@servitao:/vm/iso#
```

Figura 100. Movemos la ISO de home al directorio creado para poder instalar el sistema operativo en el disco virtual.



```
itaokvm@servitao: ~
GNU nano 7.2                      equipol.sh
#! /bin/bash

qemu-system-x86_64 --enable-kvm -name win10 -m 4G -cpu host -smp 4 -machine type=q35,accel=kvm -drive
[ 3 líneas leídas ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar^U Pegar      ^J Justificar^/ Ir a línea
```

Figura 101. Ejecutamos `# nano equipo1.sh` dentro de nuestro script ponemos

```
#!/bin/bash

qemu-system-x86_64 --enable-kvm -name ubuntu -m 4G -cpu host -smp 4 -machine type=q35,accel=kvm -drive
file=disco/disco4.qcow2,index=0,media=disk,if=ide,cache=unsafe,format=qcow2 -drive
file=iso/win10.iso,index=1,media=cdrom -net nic,macaddr=52:54:85:a3:39:f8 -net bridge,br=br0 -vga virtio -spice
port=5930,disable-ticketing=on -daemonize
```

```
root@servitao:/vm# nano equipol.sh
root@servitao:/vm# chmod +x equipol.sh
```

Figura 102. Solo damos permiso de ejecución con `# chmod +x equipo1.sh`

```
root@servitao:/vm# ./equipol.sh
```

Figura 103. Ejecutamos el script.

Comenzamos la prueba utilizando la aplicación Remote View que la descargaremos de la siguiente URL: <https://virt-manager.org/download.html> para visualizar el sistema operativo, considerando que QEMU está ejecutando el terminal y virtualizando el sistema operativo. Luego, implementaremos el protocolo SPICE para visualizar gráficamente el sistema operativo en otra computadora esto ambientado en cloud computing.



Figura 104. Ponemos la ip de la siguiente manera spice://192.168.127.133:5930.

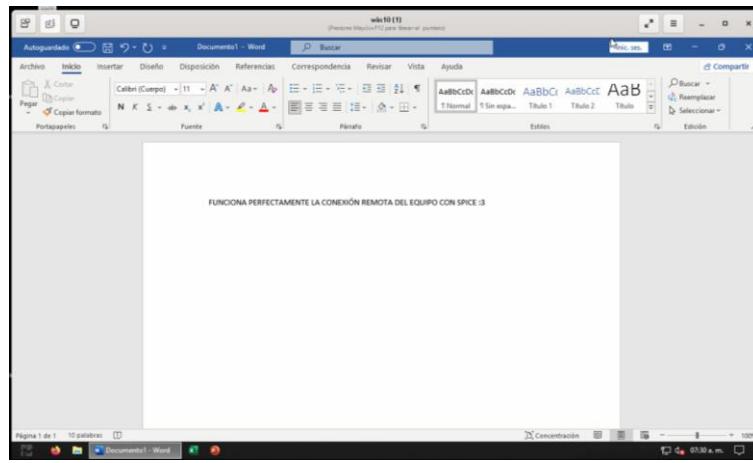
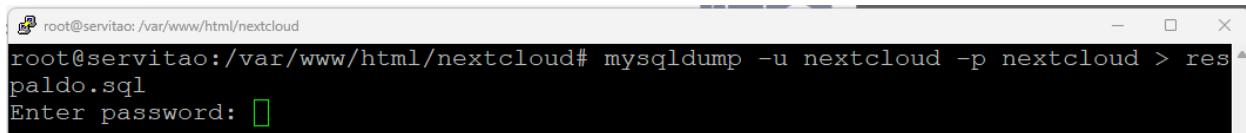


Figura 105. después ya podremos visualizar la virtualización del sistema operativo de forma remota.

ANEXO B

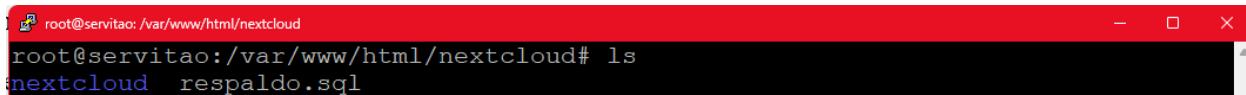
Comenzamos la migración desde Ubuntu a Debian con la transferencia de la base de datos.

Realizamos una copia de seguridad y la guardamos como un archivo .sql. Luego, en el entorno Debian, importamos el archivo .sql para restaurar la base de datos. Puedes encontrar los comandos específicos para estos pasos en la figura que se presenta a continuación.



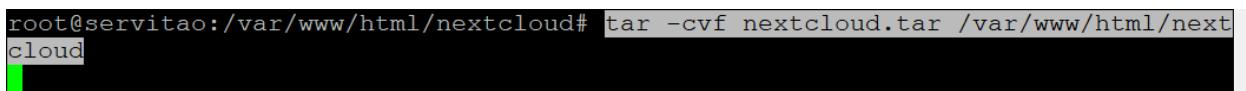
```
root@servitao:/var/www/html/nextcloud
root@servitao:/var/www/html/nextcloud# mysqldump -u nextcloud -p nextcloud > respaldo.sql
Enter password: [REDACTED]
```

Figura 106. creamos una copia de seguridad de la base de datos del servidor Ubuntu usamos el siguiente comando # mysqldump -u nextcloud -p nextcloud > respaldo.sql. después te pedirá la contraseña de la base de datos.



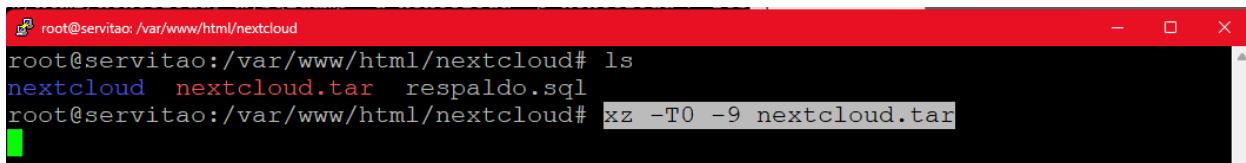
```
root@servitao:/var/www/html/nextcloud
root@servitao:/var/www/html/nextcloud# ls
nextcloud respaldo.sql
```

Figura 107. Verificamos que se haya hecho correctamente el respaldo.



```
root@servitao:/var/www/html/nextcloud# tar -cvf nextcloud.tar /var/www/html/nextcloud
```

Figura 108. Primero empaquetamos el directorio de nextcloud con el siguiente comando # tar -cvf nextcloud.tar /var/www/html/nextcloud.



```
root@servitao:/var/www/html/nextcloud
root@servitao:/var/www/html/nextcloud# ls
nextcloud nextcloud.tar respaldo.sql
root@servitao:/var/www/html/nextcloud# xz -T0 -9 nextcloud.tar
```

Figura 109. Ahora comprimimos el archivo empaquetado con un nivel 9 de compresión

```
root@servitao:/var/www/html/nextcloud# ls
nextcloud  nextcloud.tar.xz  respaldo.sql
root@servitao:/var/www/html/nextcloud# mv nextcloud.tar.xz respaldo.sql /home/itao/
root@servitao:/var/www/html/nextcloud# cd /home/itao/
root@servitao:/home/itao# ls
nextcloud.tar.xz  respaldo.sql
```

Figura 110. movemos los archivos a /home/itao con el comando # mv nextcloud.tar.xz respaldo.sql /home/itao/

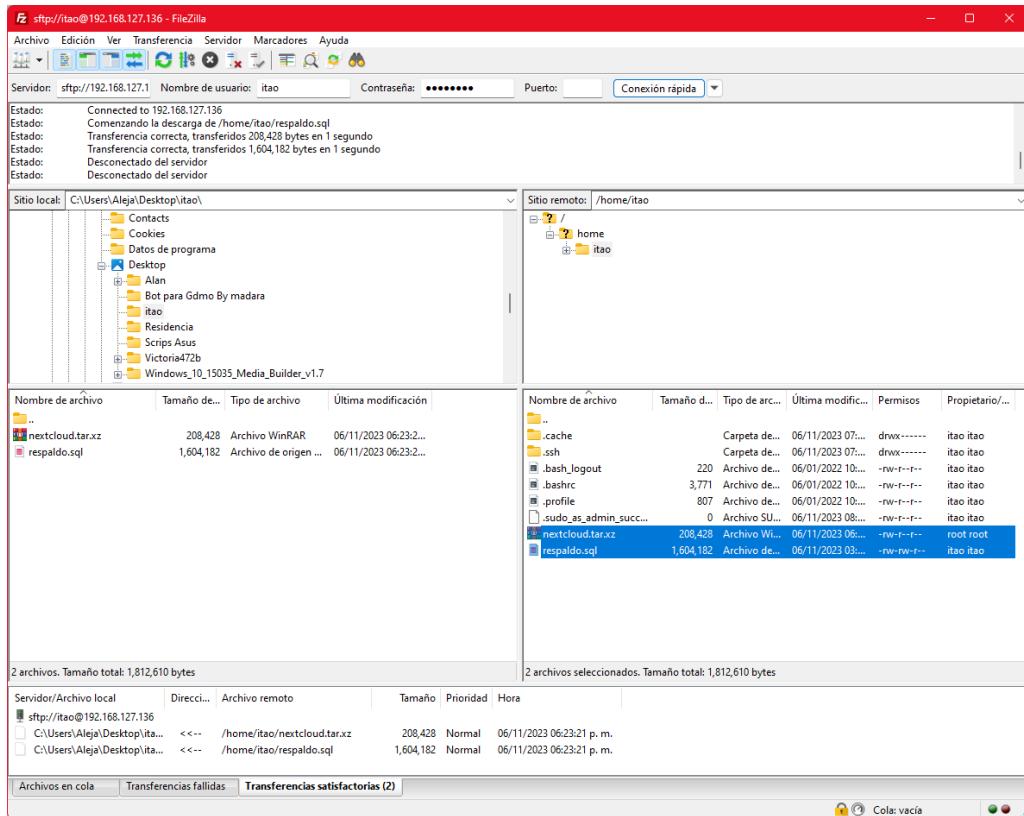
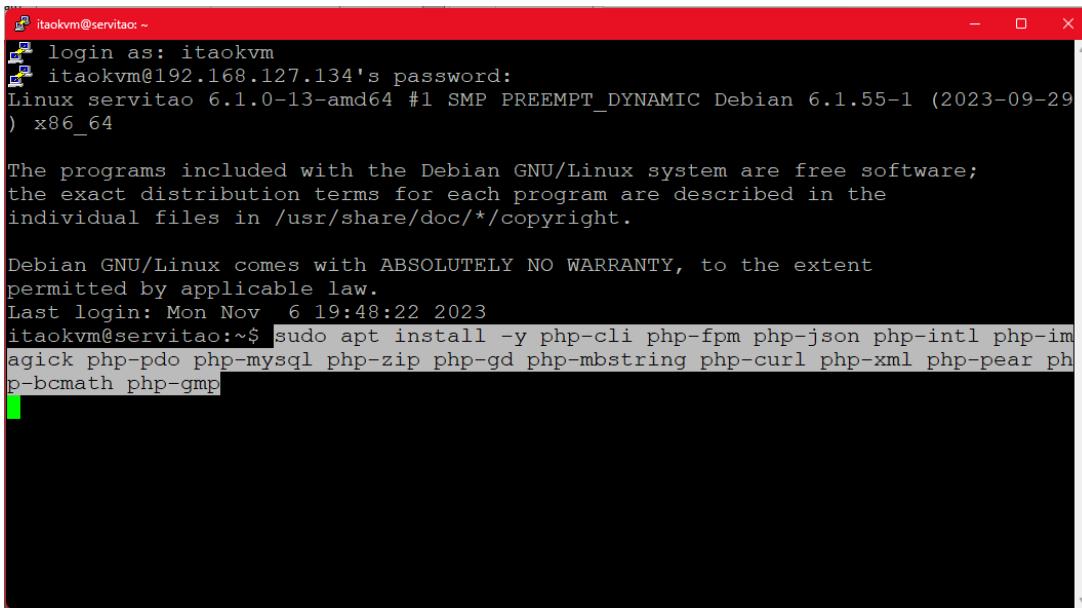


Figura 111. Paso de archivos del servidor Ubuntu a una computadora con el respaldo de nextcloud y la base de datos.

Vamos a transferir el respaldo de la base de datos de Nextcloud a Debian para la instalación completa del servidor. Una vez que el servidor Nextcloud esté en funcionamiento en Debian, instalaremos SPICE en un entorno web para Nextcloud y así ejecutar las máquinas virtuales directamente desde la plataforma.

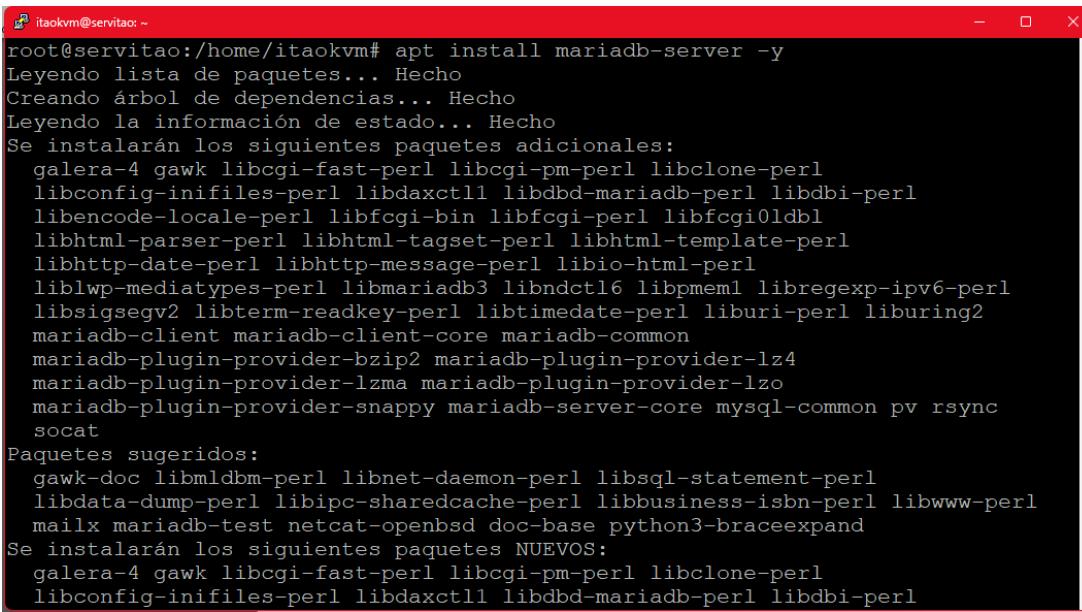


```
itaokvm@servitao: ~
login as: itaokvm
itaokvm@192.168.127.134's password:
Linux servitao 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29)
) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

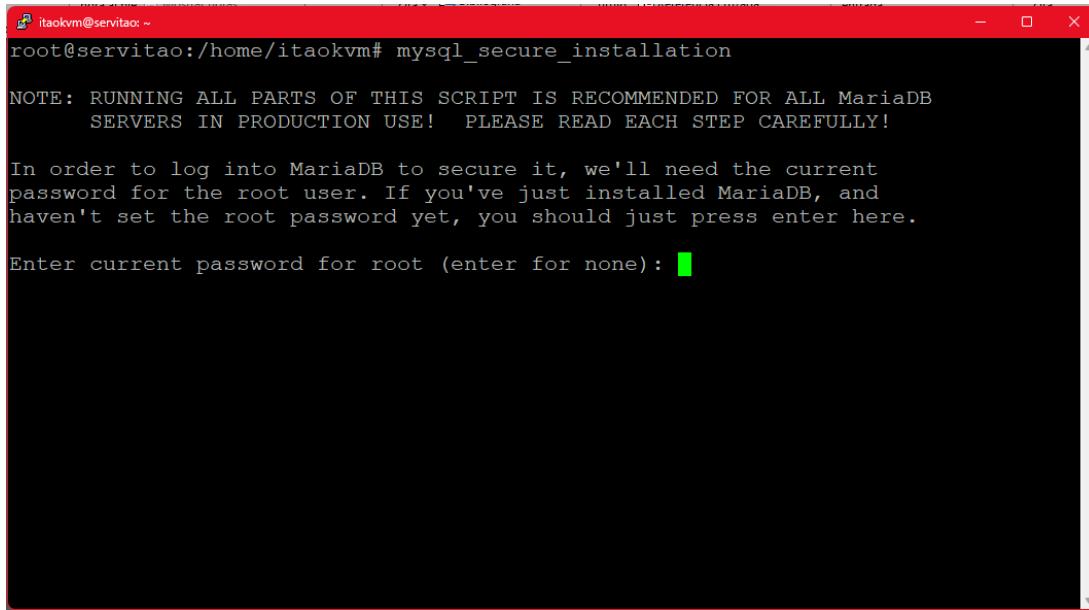
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov  6 19:48:22 2023
itaokvm@servitao:~$ sudo apt install -y php-cli php-fpm php-json php-intl php-imagick php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml php-pear php-bcmath php-gmp
```

Figura 112. instalacion de paquetes de php para el funcionamiento de nextcloud con los siguientes comandos # sudo apt install -y php-cli php-fpm php-json php-intl php-imagick php-pdo php-mysql php-zip php-gd php-mbstring php-curl php-xml php-pear php-bcmath php-gmp.



```
root@servitao:/home/itaokvm# apt install mariadb-server -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  galera-4 gawk libcgifast-perl libcgipm-perl libclone-perl
  libconfiginifiles-perl libdaxctl1 libdbd-mariadb perl libdbi-perl
  libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi01db1
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmariadb3 libndctl16 libpmem1 libregexp-ipv6-perl
  libsigsegv2 libterm-readkey-perl libtimedate-perl liburi-perl liburing2
  mariadb-client mariadb-client-core mariadb-common
  mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4
  mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo
  mariadb-plugin-provider-snappy mariadb-server-core mysql-common pv rsync
  socat
Paquetes sugeridos:
  gawk-doc libldbperl libnet-daemon-perl libsql-statement-perl
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl
  mailx mariadb-test netcat-openbsd doc-base python3-braceexpand
Se instalarán los siguientes paquetes NUEVOS:
  galera-4 gawk libcgifast-perl libcgipm-perl libclone-perl
  libconfiginifiles-perl libdaxctl1 libdbd-mariadb perl libdbi-perl
```

Figura 113. Ahora instalamos MariaDB usando el siguiente comando # apt install mariadb-server -y



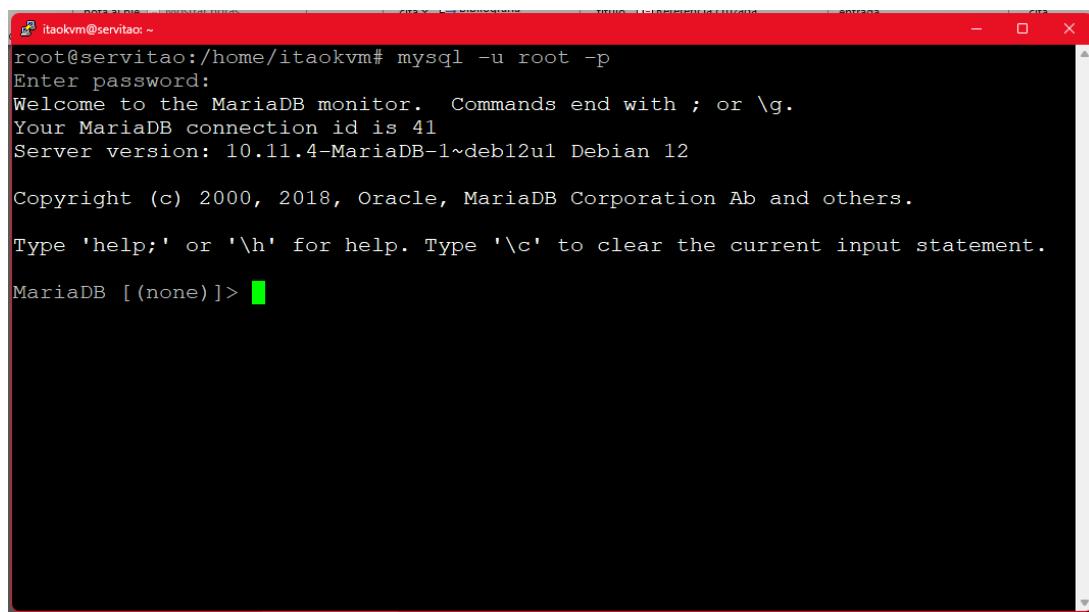
```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none): [REDACTED]
```

Figura 114. ingresamos el siguiente comando para darle seguridad a mysql # mysql_secure_installation despues le damos a todas yes e ingresamos una contraseña para volver a darle todo yes.



```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.4-MariaDB-1~deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> [REDACTED]
```

Figura 115. ahora intentamos ingresar a la base de datos con el siguiente comando y usando la contraseña que se ingresó anteriormente # mysql -u root -p.

```
itaokvm@servitao: ~
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'itaonex'@'localhost' IDENTIFIED BY 'N3XCKVMitaol23*';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> CREATE DATABASE nextcloud;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| nextcloud       |
| performance_schema |
| sys            |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]>
```

Figura 116. comenzamos a meter los comandos de la base de datos creamos el usuario con contraseña # **CREATE USER 'itaonex'@'localhost' IDENTIFIED BY 'N3XCKVMitaol23*' ;** creamos la base de datos # **CREATE DATABASE nextcloud;** verificamos si se creó la base de datos # **SHOW DATABASES;**

```
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| nextcloud       |
| performance_schema |
| sys            |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nextcloud.* TO 'itaonex'@'localhost';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> exit
Bye
root@servitao:/home/itaokvm#
```

Figura 117. asignamos permisos al usuario # **GRANT ALL PRIVILEGES ON nextcloud.* TO 'itaonex'@'localhost';** aplicamos cambios a la base de datos # **FLUSH PRIVILEGES;**

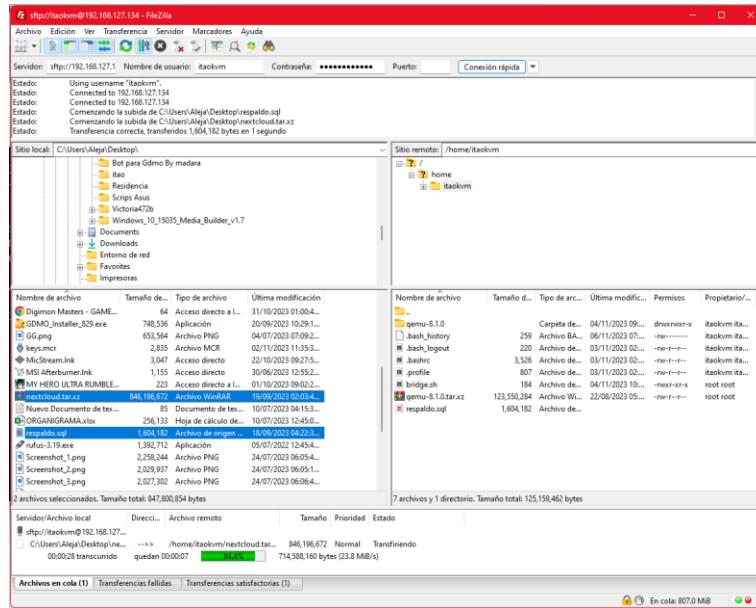


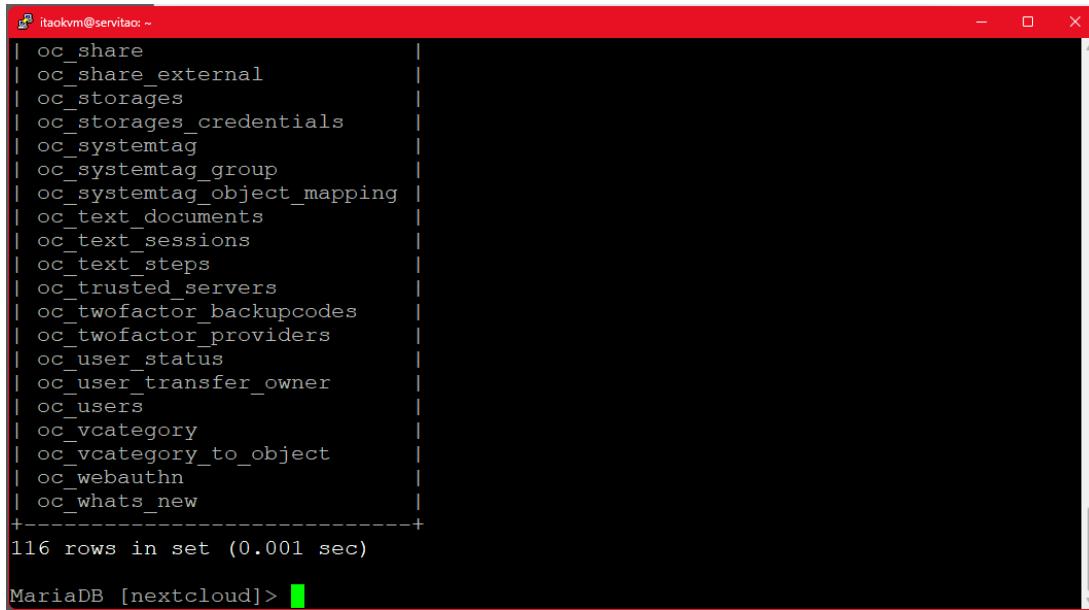
Figura 118. Pasamos los archivos respaldados a Debian

```
itaokvm@servitao: ~
  login as: itaokvm
  itaokvm@192.168.127.134's password:
Linux servitao 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29)
) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Nov  6 19:49:47 2023 from 192.168.127.1
itaokvm@servitao:~$ su
Contraseña:
root@servitao:/home/itaokvm# ls
bridge.sh nextcloud.tar.xz qemu-8.1.0 qemu-8.1.0.tar.xz respaldo.sql
root@servitao:/home/itaokvm# mysql -u itaonex -p nextcloud < /home/itaokvm/respaldo.sql
```

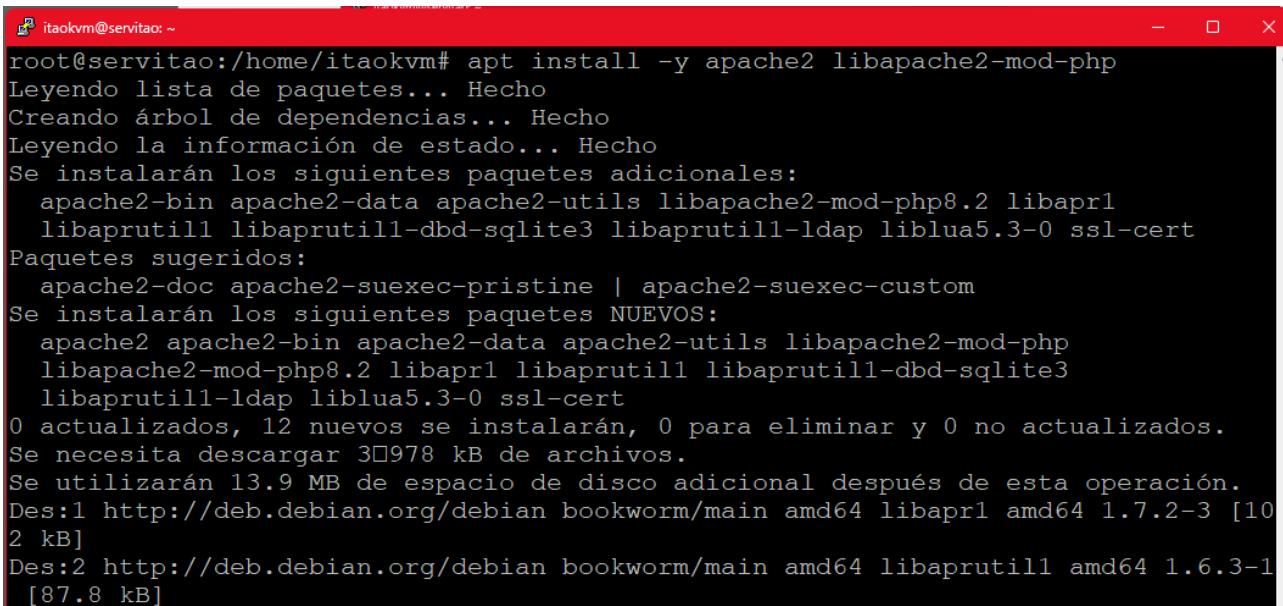
Figura 119. Importamos el respaldo de la base de datos con el siguiente comando `# mysql -u itaonex -p nextcloud < /home/itaovmkvm/respaldo.sql` después nos pedirá la contraseña que creamos del nuevo usuario.



```
itaokvm@servitao: ~
+-----+
| oc_share
| oc_share_external
| oc_storages
| oc_storages_credentials
| oc_systemtag
| oc_systemtag_group
| oc_systemtag_object_mapping
| oc_text_documents
| oc_text_sessions
| oc_text_steps
| oc_trusted_servers
| oc_twofactor_backupcodes
| oc_twofactor_providers
| oc_user_status
| oc_user_transfer_owner
| oc_users
| oc_vcATEGORY
| oc_vcATEGORY_to_object
| oc_webauthn
| oc_whats_new
+-----+
116 rows in set (0.001 sec)

MariaDB [nextcloud]>
```

Figura 120. Verificamos que estén todas las tablas, primero iniciamos en la base de datos con el siguiente comando **# mysql -u itaonex -p** ahora nos pedirá la contraseña del usuario creado, después ponemos **# show databases;** para ver las bases de datos y escogemos **nextcloud** usando **# use nextcloud** una vez dentro de la base de datos procedemos a usar **# show tables;** para ver las tablas que se importaron.



```
root@servitao:/home/itaokvm# apt install -y apache2 libapache2-mod-php
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapache2-mod-php8.2 libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php
  libapache2-mod-php8.2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.3-0 ssl-cert
0 actualizados, 12 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 30978 kB de archivos.
Se utilizarán 13.9 MB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 libapr1 amd64 1.7.2-3 [102 kB]
Des:2 http://deb.debian.org/debian bookworm/main amd64 libaprutil1 amd64 1.6.3-1 [87.8 kB]
```

Figura 121. Ahora instalamos apache 2 con el siguiente comando **# apt install -y apache2 libapache2-mod-php**



```
root@servitao:/home/itaokvm# cd /etc/apache2/sites-available/  
root@servitao:/etc/apache2/sites-available#
```

Figura 122. Una vez instalado nos movemos al siguiente directorio # cd /etc/apache2/sites-available/

```
root@servitao:/etc/apache2/sites-available# ls  
000-default.conf default-ssl.conf  
root@servitao:/etc/apache2/sites-available# cp 000-default.conf nextcloud.conf  
root@servitao:/etc/apache2/sites-available# ls  
000-default.conf default-ssl.conf nextcloud.conf  
root@servitao:/etc/apache2/sites-available#
```

Figura 123. Creamos una copia del archivo para modificar con el siguiente comando # cp 000-default.conf nextcloud.conf.

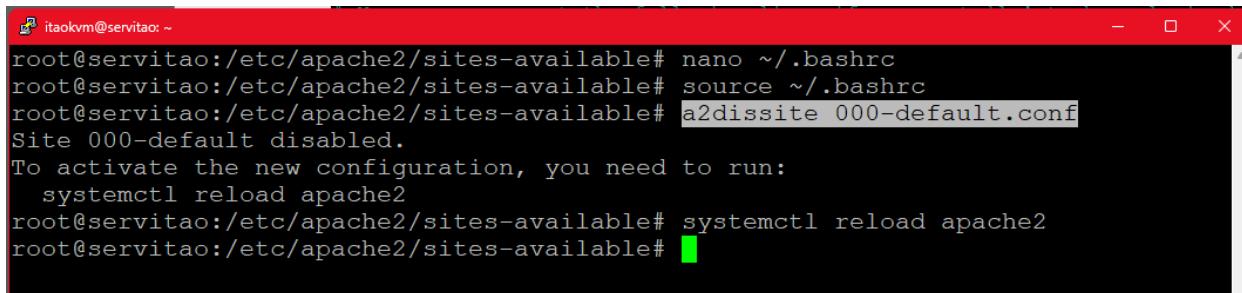
La captura de pantalla muestra una terminal en modo texto. El título de la terminal es "itaokvm@servitao: ~". La terminal ejecuta el comando "nano .bashrc" para editar el archivo .bashrc. El contenido del archivo es:

```
GNU nano 7.2 /root/.bashrc *  
# need this unless you want different defaults for root.  
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '  
# umask 022  
  
# You may uncomment the following lines if you want 'ls' to be colorized:  
# export LS_OPTIONS='--color=auto'  
# eval "$(dircolors)"  
# alias ls='ls $LS_OPTIONS'  
# alias ll='ls $LS_OPTIONS -l'  
# alias l='ls $LS_OPTIONS -la'  
  
# Some more alias to avoid making mistakes:  
# alias rm='rm -i'  
# alias cp='cp -i'  
# alias mv='mv -i'  
  
export PATH=$PATH:/home/itaokvm/qemu-8.1.0/build/qemu-bundle/opt/qemu-8/bin/  
export PATH=$PATH:/usr/sbin
```

En la parte inferior de la terminal se muestran las teclas de atajo para el editor nano:

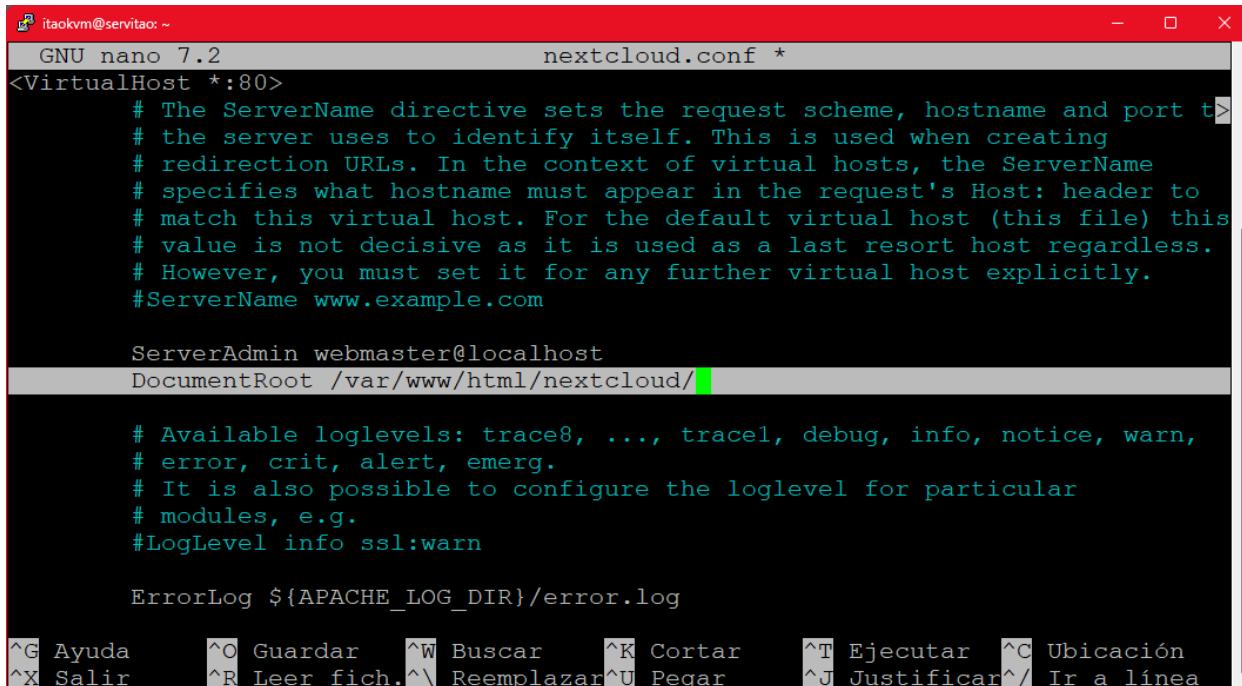
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^/ Ir a línea

Figura 124. Usamos # nano ~/.bashrc para editar el path y escribimos lo siguiente # export PATH=\$PATH:/usr/sbin y guardamos el archivo y después escribimos # source ~/.bashrc para guardar los cambios.



```
itaokvm@servitao: ~
root@servitao:/etc/apache2/sites-available# nano ~/.bashrc
root@servitao:/etc/apache2/sites-available# source ~/.bashrc
root@servitao:/etc/apache2/sites-available# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@servitao:/etc/apache2/sites-available# systemctl reload apache2
root@servitao:/etc/apache2/sites-available#
```

Figura 125. Desactivamos 000-default.conf con `# a2dissite 000-default.conf` después solo reiniciamos apache 2 con `# systemctl reload apache2`



```
GNU nano 7.2                               nextcloud.conf *
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port t>
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/nextcloud/ [highlighted]

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log

^G Ayuda      ^O Guardar     ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich.  ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea
```

Figura 126. Editamos la siguiente línea de comando `# DocumentRoot /var/www/html/nextcloud/`.



```
root@servitao:/etc/apache2/sites-available# nano nextcloud.conf
root@servitao:/etc/apache2/sites-available# a2enmod rewrite dir mime env headers

Enabling module rewrite.
Module dir already enabled
Module mime already enabled
Module env already enabled
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@servitao:/etc/apache2/sites-available# systemctl restart apache2
root@servitao:/etc/apache2/sites-available#
```

Figura 127. Ahora habilitamos los módulos con # `a2enmod rewrite dir mime env headers` después reiniciamos apache 2 #

`systemctl restart apache2`

La captura de pantalla muestra una terminal en modo nano editando el archivo `nextcloud.conf`. El contenido del archivo es:

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

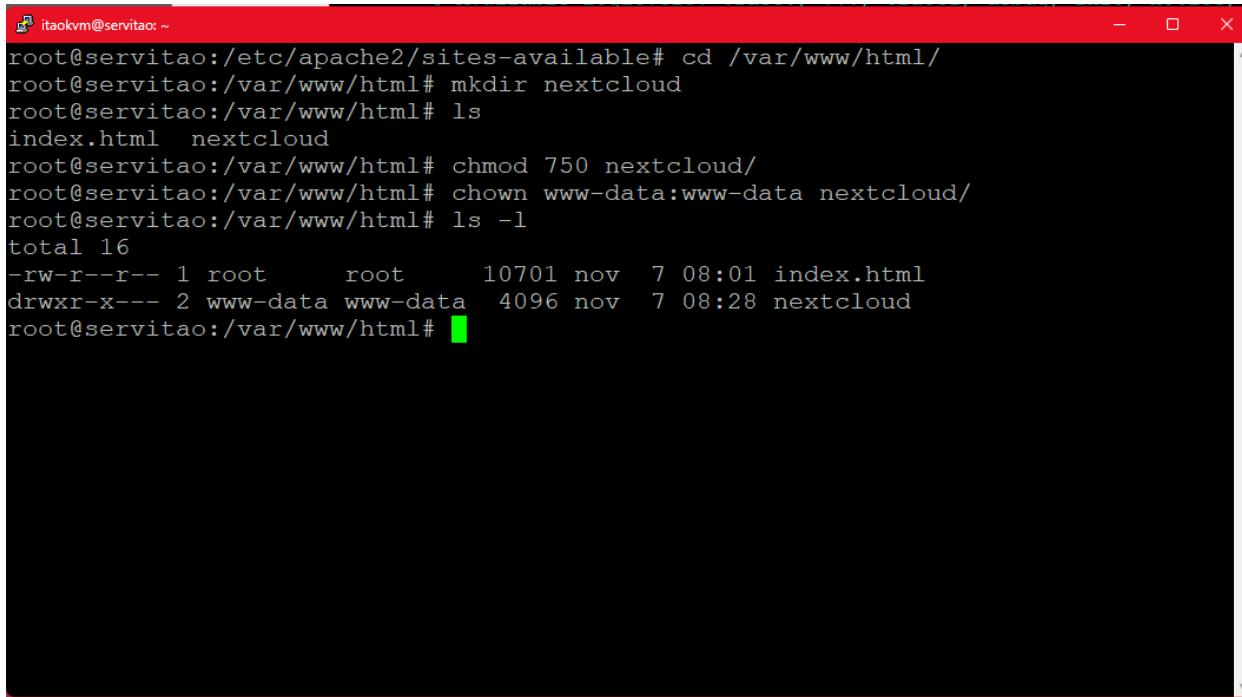

En la barra de menú de la terminal se observan las siguientes opciones:



- ^G Ayuda
- ^O Guardar
- ^W Buscar
- ^K Cortar
- ^T Ejecutar
- ^C Ubicación
- ^X Salir
- ^R Leer fich.
- ^V Reemplazar
- ^U Pegar
- ^J Justificar
- ^/ Ir a línea

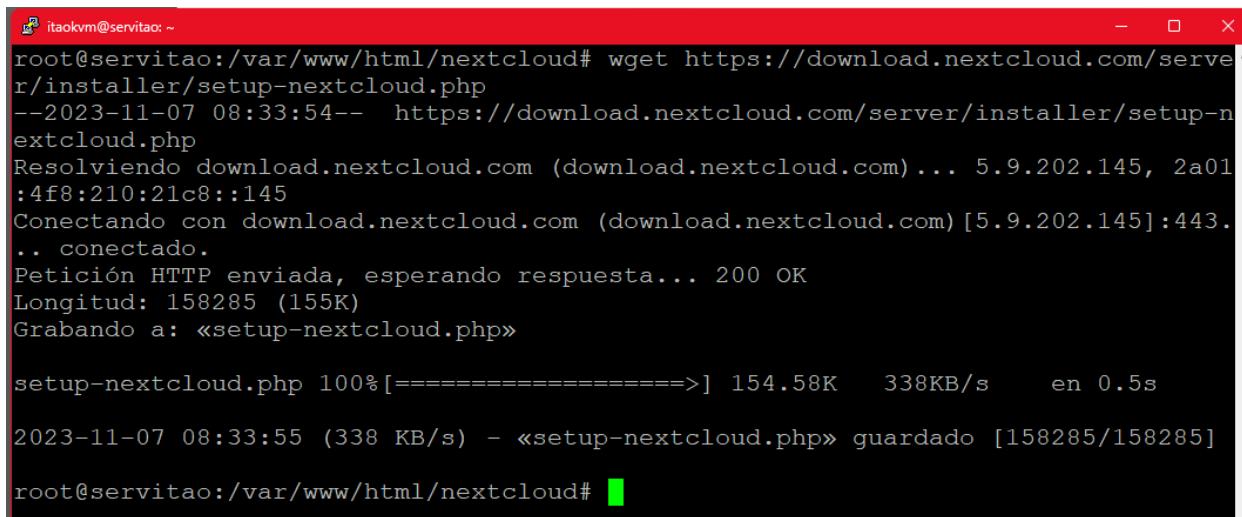
```

Figura 128. ingresamos la siguiente línea de comando en el archivo `nextcloud.conf` con # `nano nextcloud.conf`.



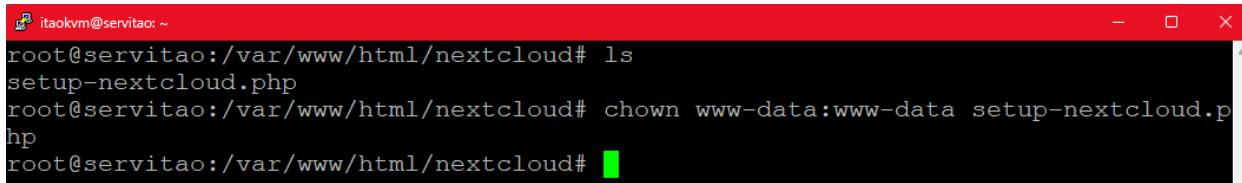
```
itaokvm@servitao: ~
root@servitao:/etc/apache2/sites-available# cd /var/www/html/
root@servitao:/var/www/html# mkdir nextcloud
root@servitao:/var/www/html# ls
index.html  nextcloud
root@servitao:/var/www/html# chmod 750 nextcloud/
root@servitao:/var/www/html# chown www-data:www-data nextcloud/
root@servitao:/var/www/html# ls -l
total 16
-rw-r--r-- 1 root      root      10701 nov  7 08:01 index.html
drwxr-x--- 2 www-data www-data  4096 nov  7 08:28 nextcloud
root@servitao:/var/www/html#
```

Figura 129. Nos movemos a la siguiente carpeta con # cd /var/www/html/ creamos un nuevo directorio # mkdir nextcloud le damos permiso 750 # chmod 750 nextcloud/ ahora asignamos un grupo # chown www-data:www-data nextcloud/.



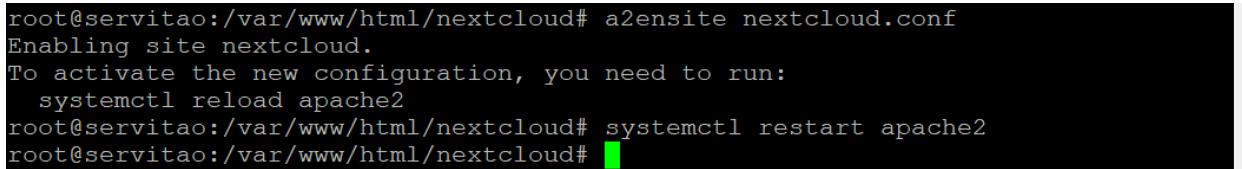
```
itaokvm@servitao: ~
root@servitao:/var/www/html/nextcloud# wget https://download.nextcloud.com/server/installer/setup-nextcloud.php
--2023-11-07 08:33:54--  https://download.nextcloud.com/server/installer/setup-nextcloud.php
Resolviendo download.nextcloud.com (download.nextcloud.com) ... 5.9.202.145, 2a01:4f8:210:21c8::145
Conectando con download.nextcloud.com (download.nextcloud.com) [5.9.202.145]:443...
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 158285 (155K)
Grabando a: «setup-nextcloud.php»
setup-nextcloud.php 100%[=====] 154.58K  338KB/s   en 0.5s
2023-11-07 08:33:55 (338 KB/s) - «setup-nextcloud.php» guardado [158285/158285]
root@servitao:/var/www/html/nextcloud#
```

Figura 130. Descargamos el repositorio usando # wget <https://download.nextcloud.com/server/installer/setup-nextcloud.php>



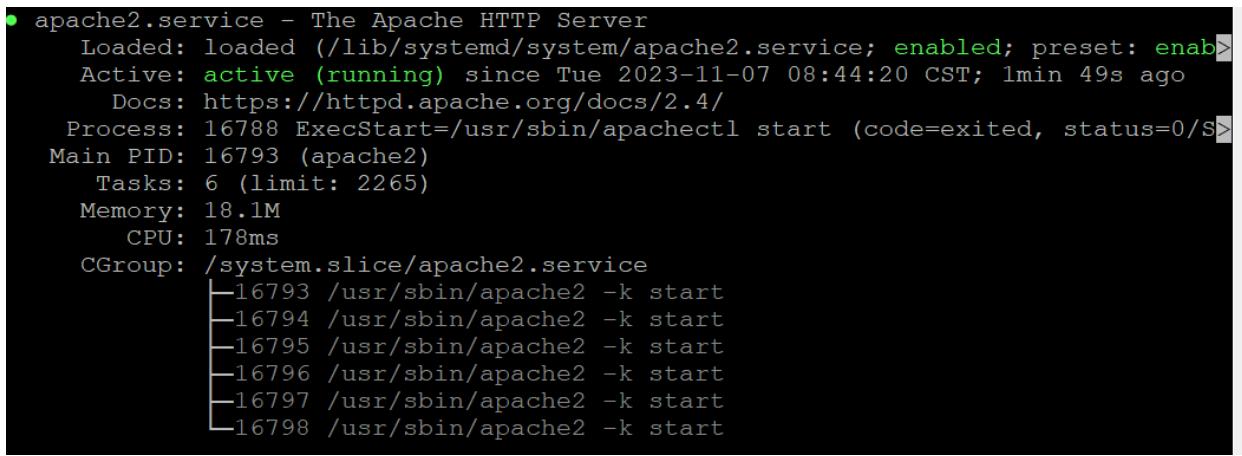
```
root@servitao:/var/www/html/nextcloud# ls
setup-nextcloud.php
root@servitao:/var/www/html/nextcloud# chown www-data:www-data setup-nextcloud.php
root@servitao:/var/www/html/nextcloud#
```

Figura 131. Asignamos un grupo al repositorio descargado



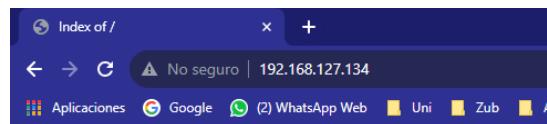
```
root@servitao:/var/www/html/nextcloud# a2ensite nextcloud.conf
Enabling site nextcloud.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@servitao:/var/www/html/nextcloud# systemctl restart apache2
root@servitao:/var/www/html/nextcloud#
```

Figura 132. Ahora activamos el `nextcloud.conf` # `a2ensite nextcloud.conf` procedemos a reiniciar apache 2 # `systemctl restart apache2`.



```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2023-11-07 08:44:20 CST; 1min 49s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 16788 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
 Main PID: 16793 (apache2)
    Tasks: 6 (limit: 2265)
   Memory: 18.1M
      CPU: 178ms
     CGroup: /system.slice/apache2.service
             └─16793 /usr/sbin/apache2 -k start
                  ├─16794 /usr/sbin/apache2 -k start
                  ├─16795 /usr/sbin/apache2 -k start
                  ├─16796 /usr/sbin/apache2 -k start
                  ├─16797 /usr/sbin/apache2 -k start
                  └─16798 /usr/sbin/apache2 -k start
```

Figura 133. Verificamos que el estatus de apache 2 esté funcionando correctamente # `systemctl status apache2`



Index of /

Name	Last modified	Size	Description
setup-nextcloud.php	2021-11-30 08:53	155K	

Apache/2.4.57 (Debian) Server at 192.168.127.134 Port 80

Figura 134. Entramos en el navegador usando la IP del servidor y abrimos el *setup-nextcloud.php*

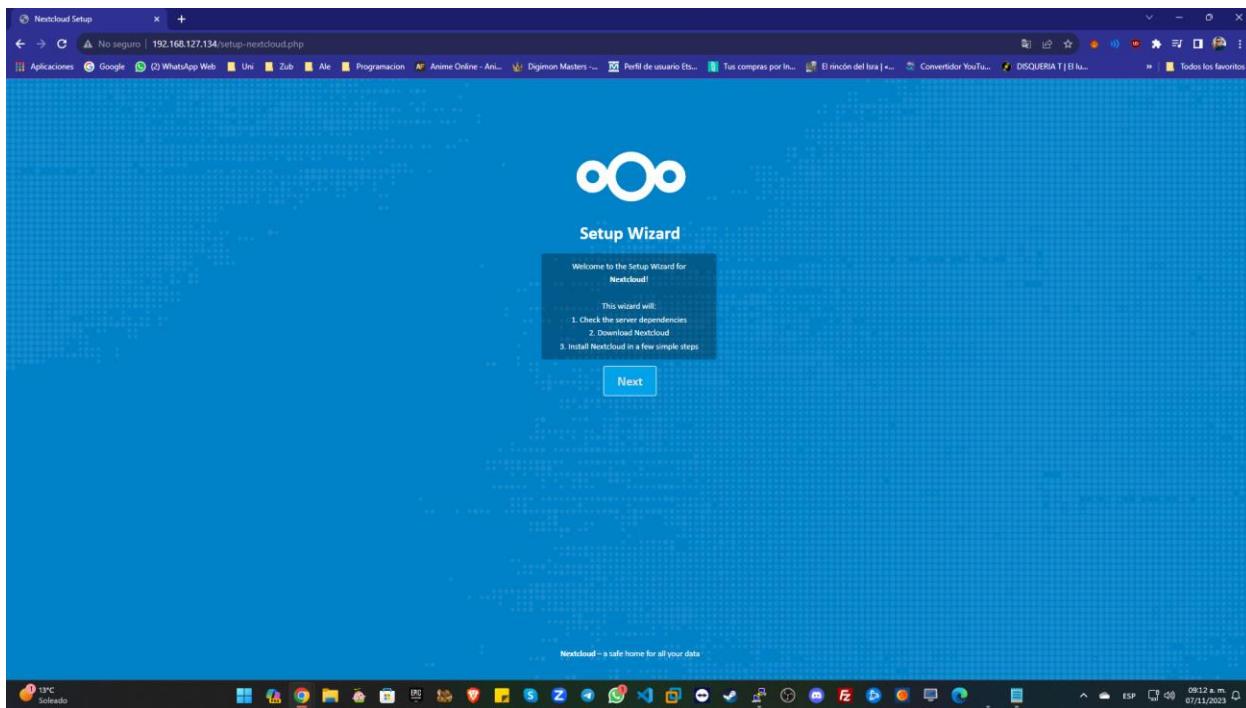
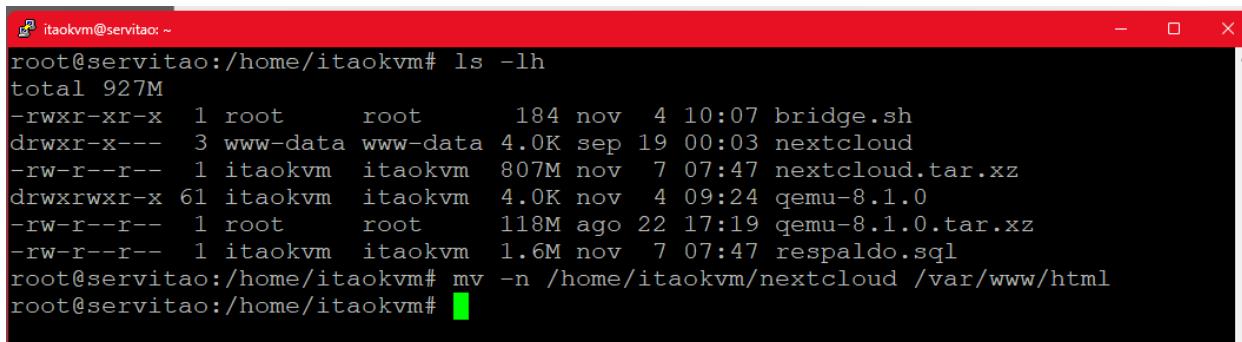


Figura 135. Configuramos el nextcloud para ahora si poder remplazar los archivos necesarios.

```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# ls
bridge.sh nextcloud.tar.xz qemu-8.1.0 qemu-8.1.0.tar.xz respaldo.sql
root@servitao:/home/itaokvm# tar Jvxf nextcloud.tar.xz
```

Figura 136. Descomprimimos el archivo *nextcloud.tar.xz* # *tar Jvxf nextcloud.tar.xz*



```
itaokvm@servitao: ~
root@servitao:/home/itaokvm# ls -lh
total 927M
-rwxr-xr-x 1 root      root      184 nov  4 10:07 bridge.sh
drwxr-x--- 3 www-data  www-data  4.0K sep 19 00:03 nextcloud
-rw-r--r-- 1 itaokvm   itaokvm  807M nov  7 07:47 nextcloud.tar.xz
drwxrwxr-x 61 itaokvm  itaokvm  4.0K nov  4 09:24 qemu-8.1.0
-rw-r--r-- 1 root      root     118M ago 22 17:19 qemu-8.1.0.tar.xz
-rw-r--r-- 1 itaokvm   itaokvm  1.6M nov  7 07:47 respaldo.sql
root@servitao:/home/itaokvm# mv -n /home/itaokvm/nextcloud /var/www/html
root@servitao:/home/itaokvm#
```

Figura 137. primero verificamos que el archivo se descomprimió usando # **ls -lh** después movemos el contenido descomprimido usando # **mv -n /home/itaovmkvm/nextcloud /var/www/html**.



This version of Nextcloud is not compatible with PHP \geq 8.2.
You are currently running 8.2.7.

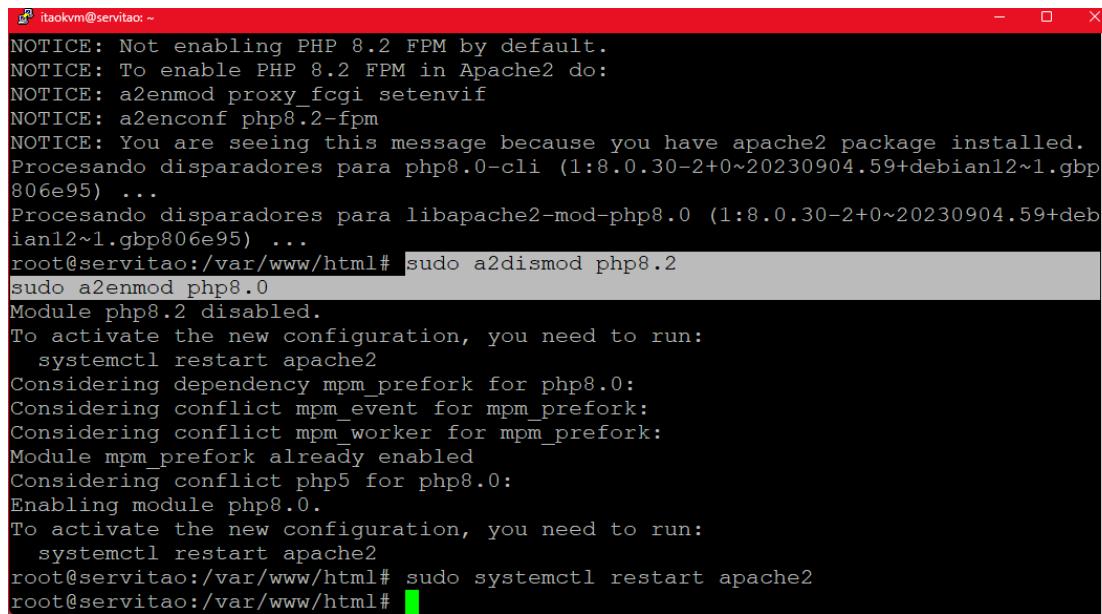
Figura 138. Nos pedirá un PHP más antiguo así que procedemos a instalarlo

```
itaokvm@servitao:~  
root@servitao:/var/www/html# sudo apt install lsb-release ca-certificates apt-tr  
ansport-https software-properties-common -y  
sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gp  
g  
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" | sudo tee /et  
c/apt/sources.list.d/php.list  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
lsb-release ya está en su versión más reciente (12.0-1).  
fijado lsb-release como instalado manualmente.  
ca-certificates ya está en su versión más reciente (20230311).  
Se instalarán los siguientes paquetes adicionales:  
 appstream gir1.2-packagekitglib-1.0 libappstream4 libduktape207  
 libpackagekit-glib2-18 libpolkit-agent-1-0 libpolkit-gobject-1-0  
 libstemmer0d libxml2-2 libyaml-0-2 packagekit packagekit-tools polkitd  
 python3-blinker python3-cffi-backend python3-cryptography python3-dbus  
 python3-distro python3-gi python3-jwt python3-lazr.restfulclient  
 python3-lazr.uri python3-oauthlib python3-software-properties  
 python3-wadllib  
Paquetes sugeridos:  
 apt-config-icons polkitd-pkla python-blinker-doc python-cryptography-doc  
 python3-cryptography-vectors python-dbus-doc python3-crypto  
Se instalarán los siguientes paquetes NUEVOS:
```

Figura 139. Descargamos el repositorio de PHP que necesitamos

```
root@servitao:/var/www/html# sudo apt update
sudo apt install php8.0 php8.0-common php8.0-mysql php8.0-xml php8.0-xmlrpc php8
.0-curl php8.0-gd php8.0-imagick php8.0-cli php8.0-dev php8.0-imap php8.0-mbstri
ng php8.0-opcache php8.0-soap php8.0-zip php8.0-intl -y
Des:1 https://packages.sury.org/php bookworm InRelease [70539 B]
Obj:2 http://deb.debian.org/debian bookworm InRelease
Des:3 http://security.debian.org/debian-security bookworm-security InRelease [48
.0 kB]
Des:4 http://deb.debian.org/debian bookworm-updates InRelease [52.1 kB]
Des:5 https://packages.sury.org/php bookworm/main amd64 Packages [208 kB]
Des:6 http://deb.debian.org/debian bookworm/main amd64 DEP-11 Metadata [40492 kB]
]
Des:7 http://deb.debian.org/debian bookworm/non-free-firmware amd64 DEP-11 Metad
ata [15.5 kB]
Des:8 http://security.debian.org/debian-security bookworm-security/main Sources
[57.9 kB]
Des:9 http://security.debian.org/debian-security bookworm-security/main amd64 Pa
ckages [96.0 kB]
Des:10 http://security.debian.org/debian-security bookworm-security/main Transla
tion-en [54.7 kB]
0% [ 6 Components-amd64 store 0 B] [
```

Figura 140. Actualizamos e instalamos la versión 8.0



```

itaokvm@servitao: ~
NOTICE: Not enabling PHP 8.2 FPM by default.
NOTICE: To enable PHP 8.2 FPM in Apache2 do:
NOTICE: a2enmod proxy_fcgi setenvif
NOTICE: a2enconf php8.2-fpm
NOTICE: You are seeing this message because you have apache2 package installed.
Procesando disparadores para php8.0-cli (1:8.0.30-2+0~20230904.59+debian12~1.gbp
806e95) ...
Procesando disparadores para libapache2-mod-php8.0 (1:8.0.30-2+0~20230904.59+deb
ian12~1.gbp806e95) ...
root@servitao:/var/www/html# sudo a2dismod php8.2
sudo a2enmod php8.0
Module php8.2 disabled.
To activate the new configuration, you need to run:
    systemctl restart apache2
Considering dependency mpm_prefork for php8.0:
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Module mpm_prefork already enabled
Considering conflict php5 for php8.0:
Enabling module php8.0.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@servitao:/var/www/html# sudo systemctl restart apache2
root@servitao:/var/www/html# 

```

Figura 141. Desactivamos la versión anterior # `sudo a2dismod php8.2` activamos la nueva # `sudo a2enmod php8.0` procedemos a reiniciar apache 2 # `sudo systemctl restart apache2`

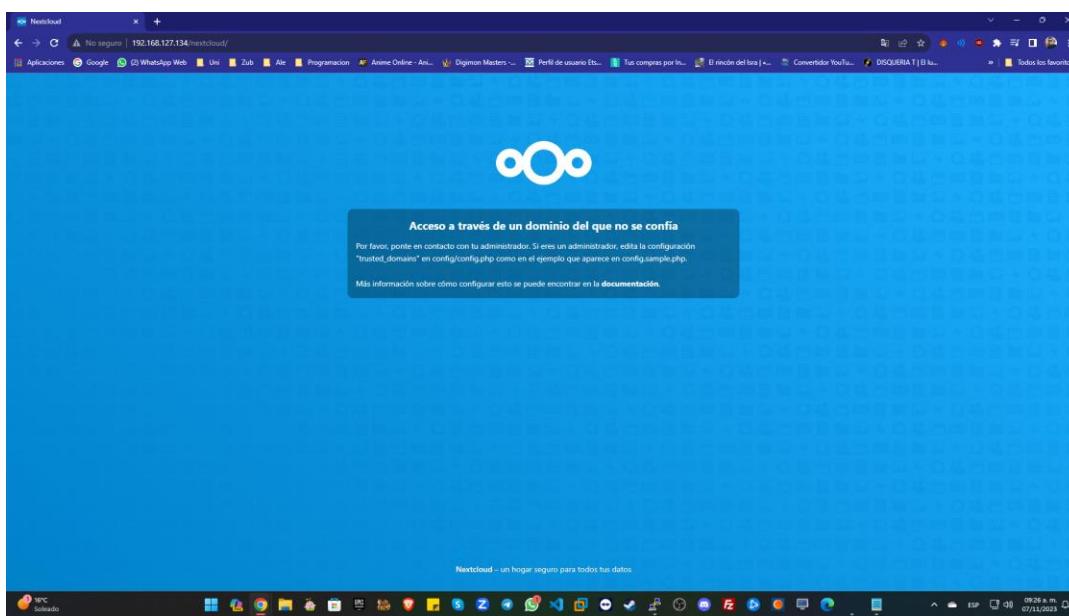
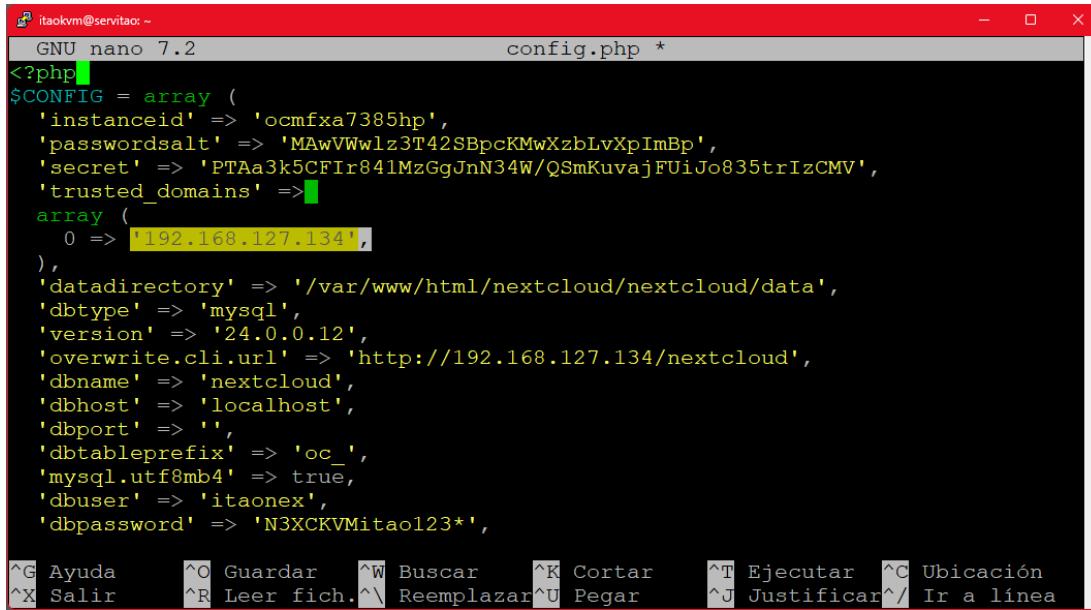


Figura 142. ahora solo nos falta configurar la IP en el nextcloud para terminar



```

itaokvm@servitao: ~
GNU nano 7.2 config.php *
<?php
$CONFIG = array (
  'instanceid' => 'ocmfxa7385hp',
  'passwordsalt' => 'MAwVWwlz3T42SBpcKMwXzbLvXpImBp',
  'secret' => 'PTAAa3k5CFIr841MzGgJnN34W/QSmKuvajFUiJo835trIzCMV',
  'trusted_domains' =>
  array (
    0 => '192.168.127.134',
  ),
  'datadirectory' => '/var/www/html/nextcloud/nextcloud/data',
  'dbtype' => 'mysql',
  'version' => '24.0.0.12',
  'overwrite.cli.url' => 'http://192.168.127.134/nextcloud',
  'dbname' => 'nextcloud',
  'dbhost' => 'localhost',
  'dbport' => '',
  'dbtableprefix' => 'oc_',
  'mysql.utf8mb4' => true,
  'dbuser' => 'itaonex',
  'dbpassword' => 'N3XCKVMitaol23*',
);

^G Ayuda      ^O Guardar   ^W Buscar   ^K Cortar   ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich.^` Reemplazar^U Pegar   ^J Justificar^/ Ir a línea

```

Figura 143 ahora solo configuraremos la siguiente ip usando # nano /var/www/html/nextcloud/nextcloud/config/config.php.

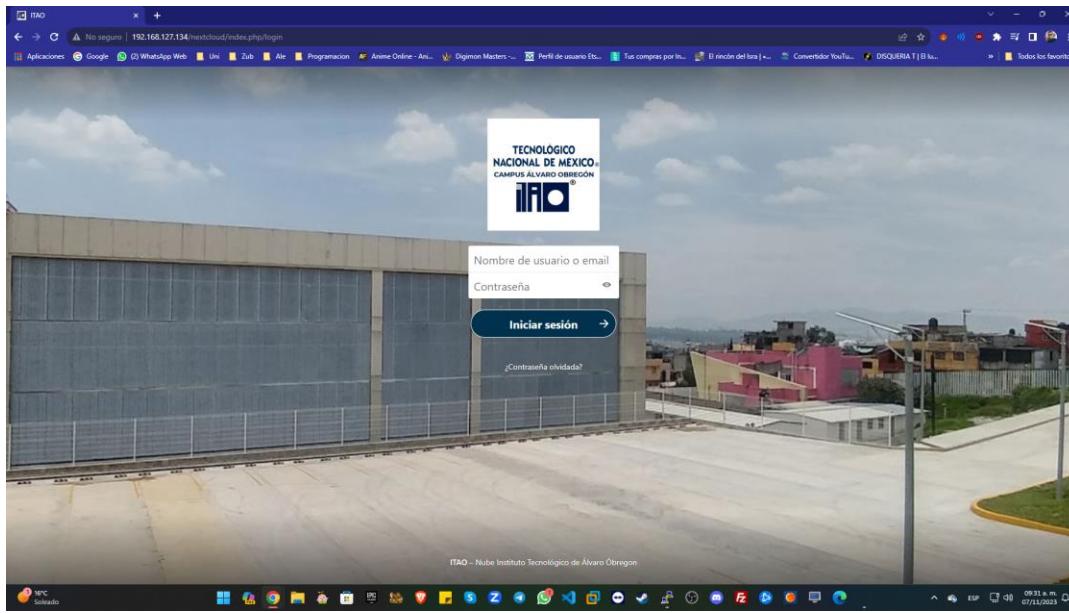
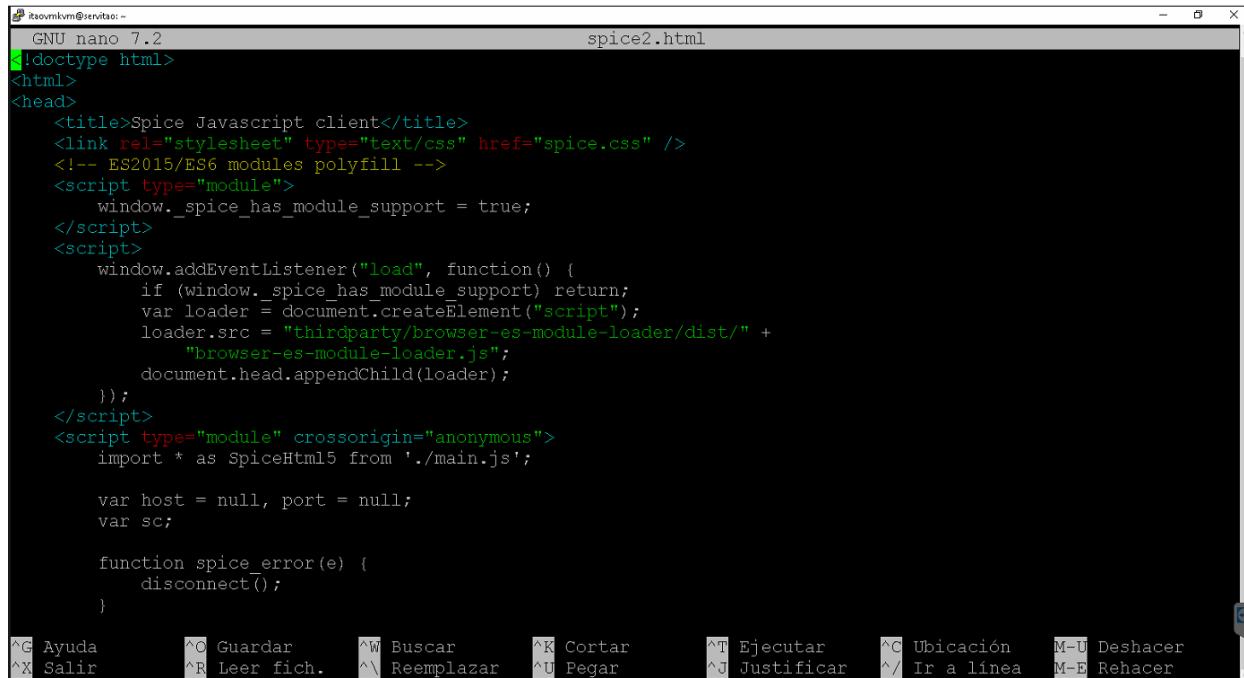


Figura 144. Ahora ya podremos tener ejecutando correctamente el nextcloud.



```

itavmkm@servitao: ~
GNU nano 7.2                               spice2.html

<!doctype html>
<html>
<head>
    <title>Spice Javascript client</title>
    <link rel="stylesheet" type="text/css" href="spice.css" />
    <!-- ES2015/ES6 modules polyfill -->
    <script type="module">
        window._spice_has_module_support = true;
    </script>
    <script>
        window.addEventListener("load", function() {
            if (window.spice_has_module_support) return;
            var loader = document.createElement("script");
            loader.src = "thirdparty/browser-es-module-loader/dist/" +
                "browser-es-module-loader.js";
            document.head.appendChild(loader);
        });
    </script>
    <script type="module" crossorigin="anonymous">
        import * as SpiceHtml5 from './main.js';

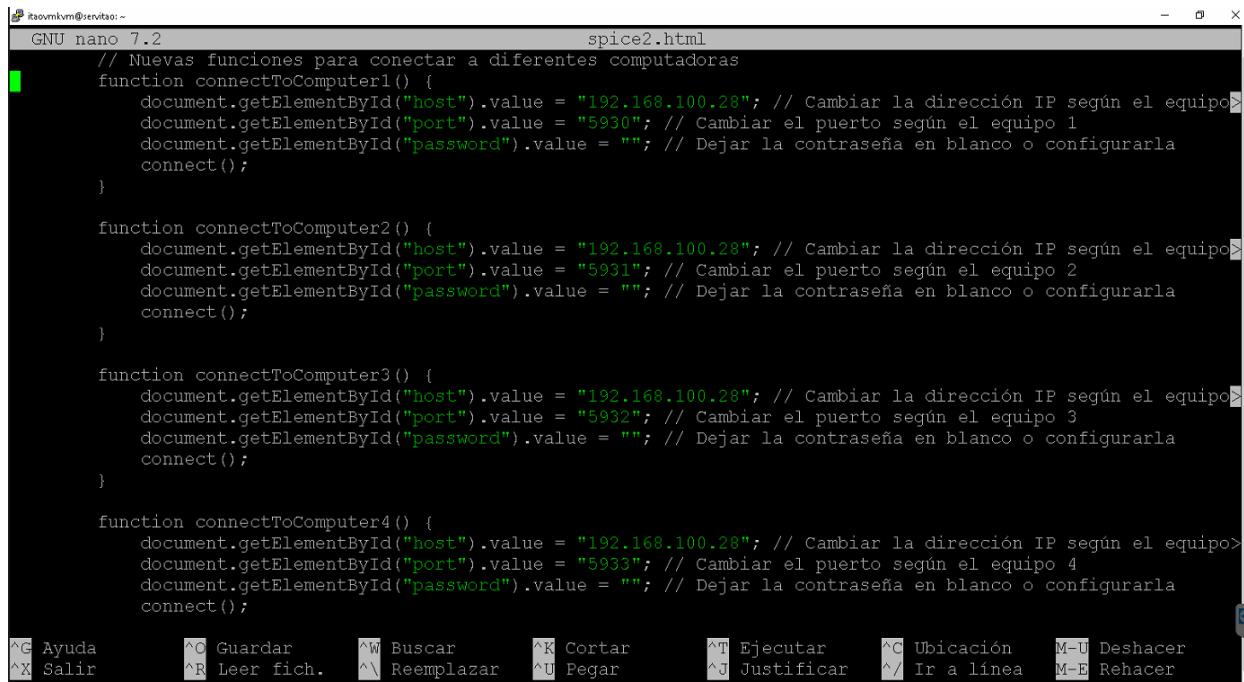
        var host = null, port = null;
        var sc;

        function spice_error(e) {
            disconnect();
        }
    </script>

```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación M-U Deshacer
 ^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^/ Ir a linea M-E Rehacer

Figura 145. Configuramos nuestro nuevo código de ápice modificado



```

itavmkm@servitao: ~
GNU nano 7.2                               spice2.html

// Nuevas funciones para conectar a diferentes computadoras
function connectToComputer1() {
    document.getElementById("host").value = "192.168.100.28"; // Cambiar la dirección IP según el equipo
    document.getElementById("port").value = "5930"; // Cambiar el puerto según el equipo 1
    document.getElementById("password").value = ""; // Dejar la contraseña en blanco o configurarla
    connect();
}

function connectToComputer2() {
    document.getElementById("host").value = "192.168.100.28"; // Cambiar la dirección IP según el equipo
    document.getElementById("port").value = "5931"; // Cambiar el puerto según el equipo 2
    document.getElementById("password").value = ""; // Dejar la contraseña en blanco o configurarla
    connect();
}

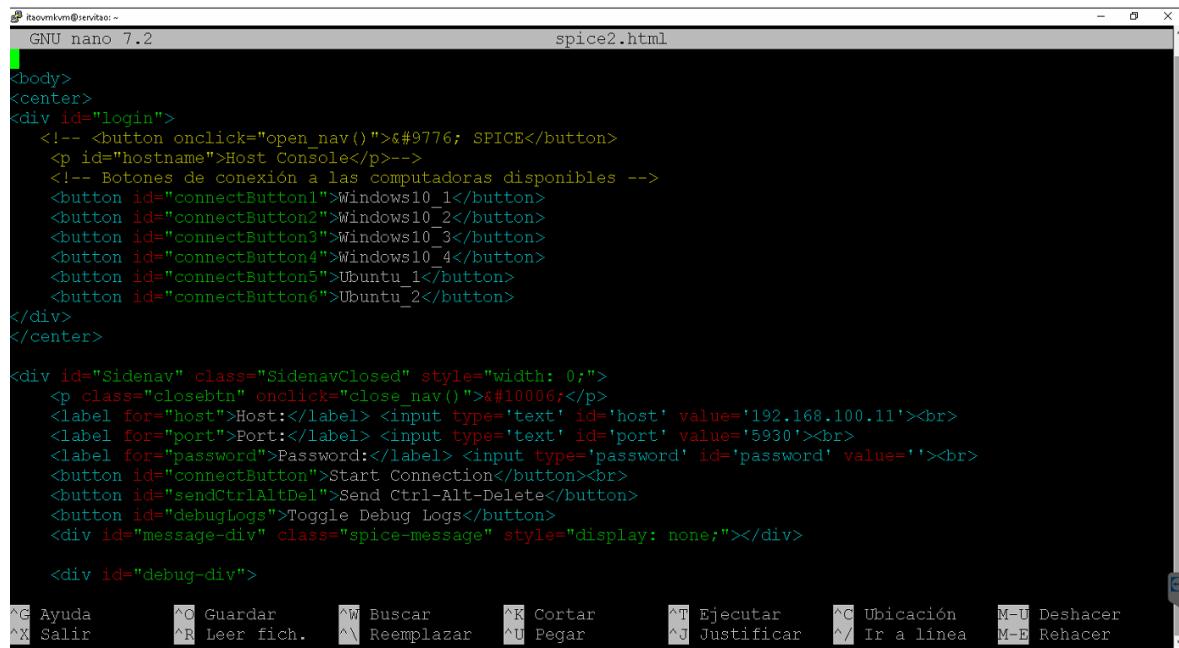
function connectToComputer3() {
    document.getElementById("host").value = "192.168.100.28"; // Cambiar la dirección IP según el equipo
    document.getElementById("port").value = "5932"; // Cambiar el puerto según el equipo 3
    document.getElementById("password").value = ""; // Dejar la contraseña en blanco o configurarla
    connect();
}

function connectToComputer4() {
    document.getElementById("host").value = "192.168.100.28"; // Cambiar la dirección IP según el equipo
    document.getElementById("port").value = "5933"; // Cambiar el puerto según el equipo 4
    document.getElementById("password").value = ""; // Dejar la contraseña en blanco o configurarla
    connect();
}

```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación M-U Deshacer
 ^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^/ Ir a linea M-E Rehacer

Figura 146. Agregamos las ip de ejecución con sus puertos para visualizar las máquinas virtuales



The screenshot shows a terminal window titled "spice2.html" with the command "GNU nano 7.2" at the top. The window contains the source code for a web page. The code includes HTML for a login form and a sidebar, and JavaScript for handling connections to various hosts. At the bottom, there is a keybinding legend for navigating the terminal.

```
itapvmlvm@servitao:~$ GNU nano 7.2
spice2.html

<body>
<center>
<div id="login">
  <!-- <button onclick="open_nav()">SPICE</button>
  <p id="hostname">Host Console</p>-->
  <!-- Botones de conexión a las computadoras disponibles -->
  <button id="connectButton1">Windows10_1</button>
  <button id="connectButton2">Windows10_2</button>
  <button id="connectButton3">Windows10_3</button>
  <button id="connectButton4">Windows10_4</button>
  <button id="connectButton5">Ubuntu_1</button>
  <button id="connectButton6">Ubuntu_2</button>
</div>
</center>

<div id="Sidenav" class="SidenavClosed" style="width: 0;">
  <p class="closebtn" onclick="close_nav()">&#10006;</p>
  <label for="host">Host:</label> <input type='text' id='host' value='192.168.100.11'><br>
  <label for="port">Port:</label> <input type='text' id='port' value='5930'><br>
  <label for="password">Password:</label> <input type='password' id='password' value=''/><br>
  <button id="connectButton">Start Connection</button><br>
  <button id="sendCtrlAltDel">Send Ctrl-Alt-Delete</button>
  <button id="debuglogs">Toggle Debug Logs</button>
  <div id="message-div" class="spice-message" style="display: none;"></div>
</div>
<div id="debug-div">

  ^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación      M-U Deshacer
  ^X Salir      ^R Leer fich.  ^\ Reemplazar   ^U Pegar       ^J Justificar   ^/ Ir a línea  M-E Rehacer

</div>
```

Figura 147. Agregamos botones para cada máquina que se va ejecutar

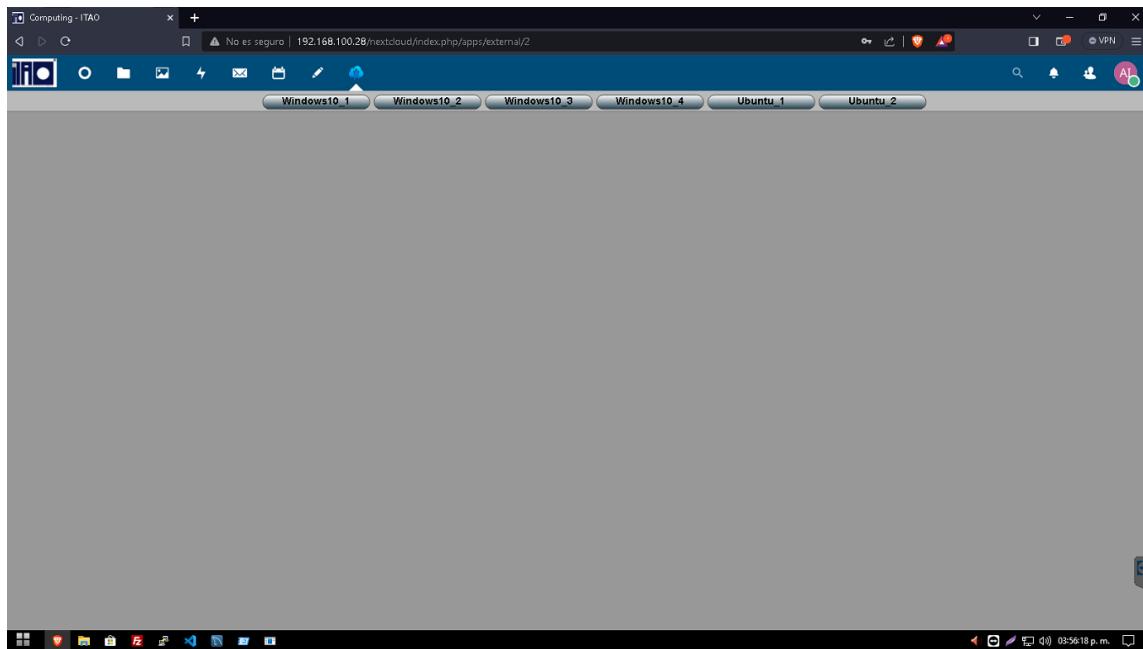


Figura 148. Se vera de la siguiente forma para poder visualizar las maquinas

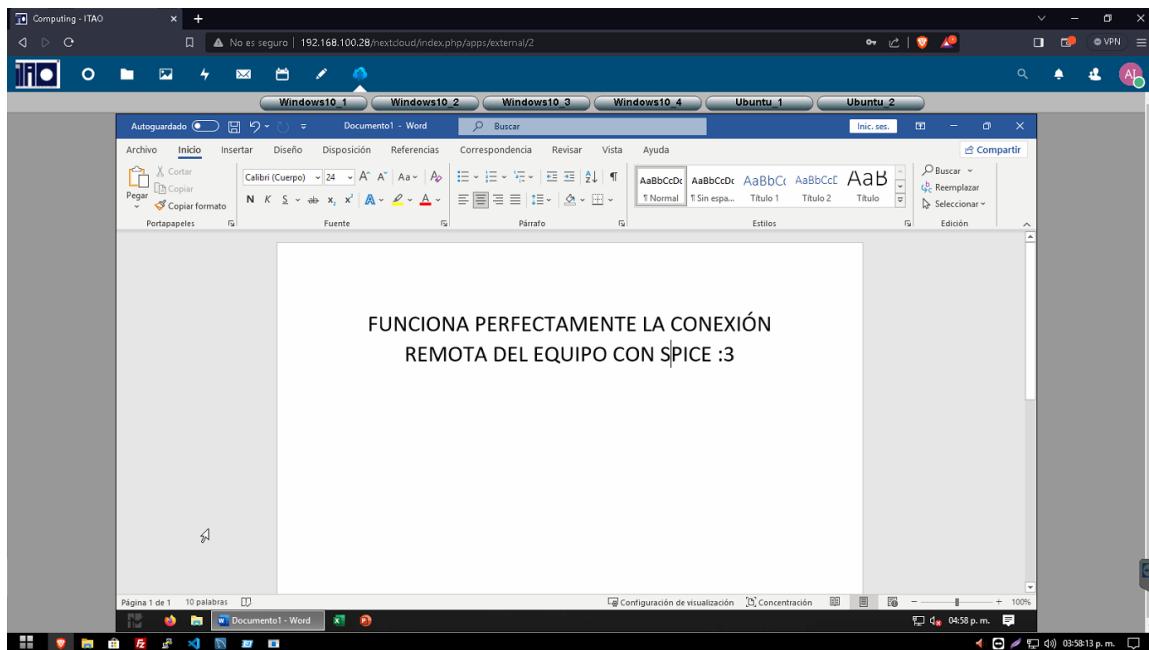


Figura 149. Esta es una maquina Windows donde se ejecuta Word desde el navegador

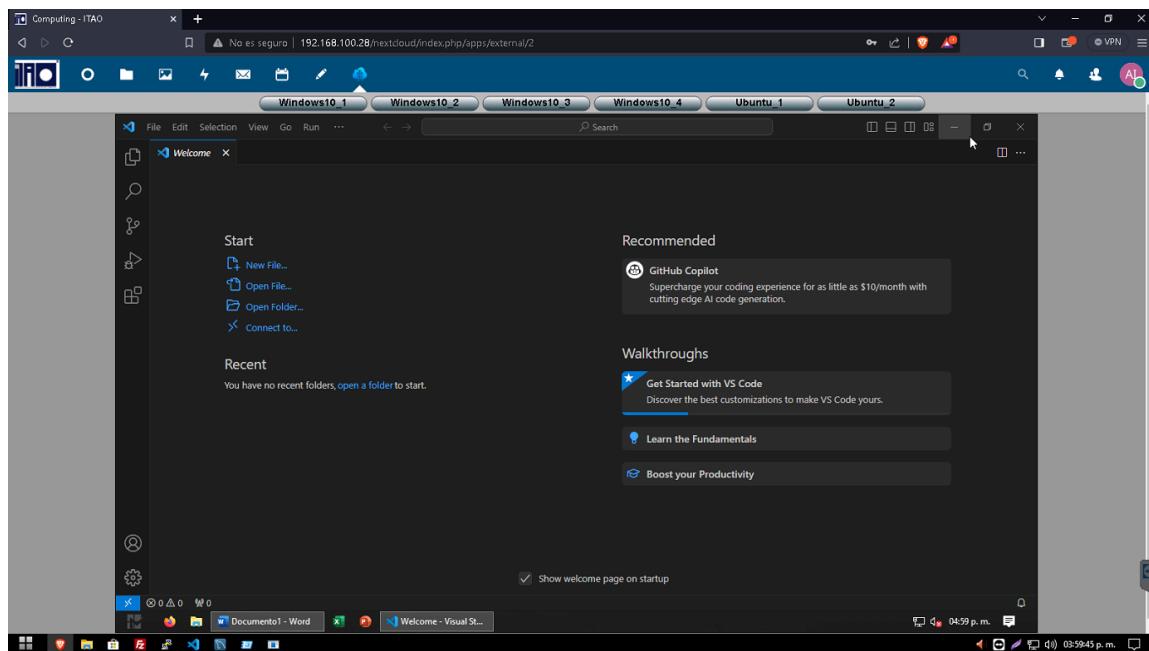


Figura 150. Aquí se está ejecutando Visual Code

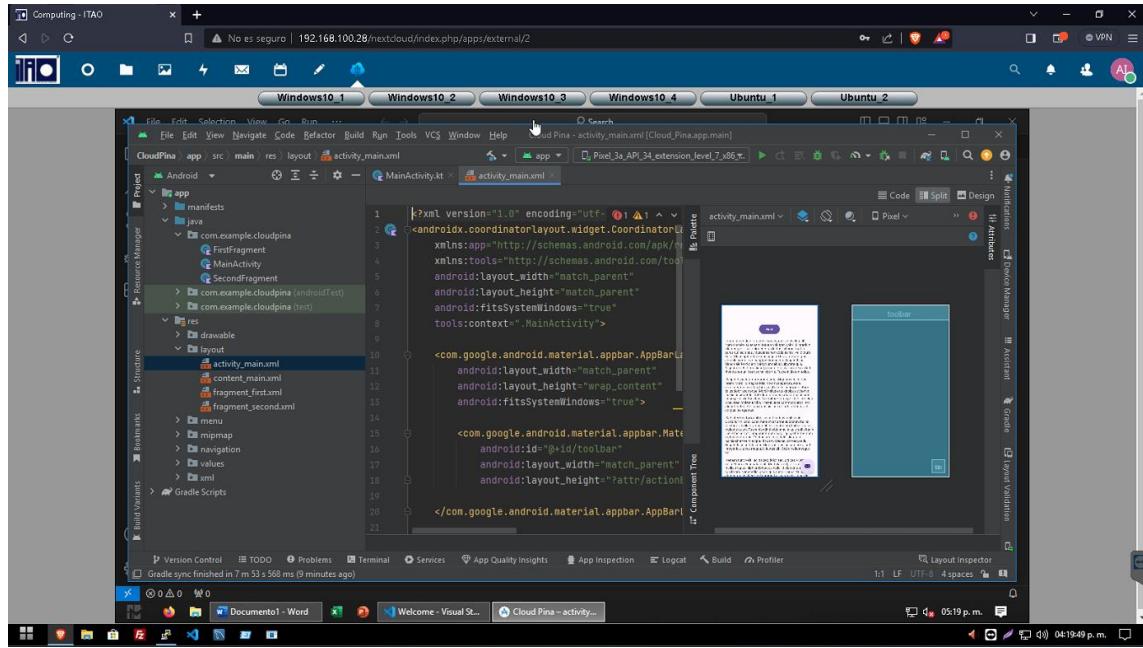


Figura 151. Por último, ejecutamos Android Studio