

Bloque 1: Instalación, administración básica y configuración del arranque.

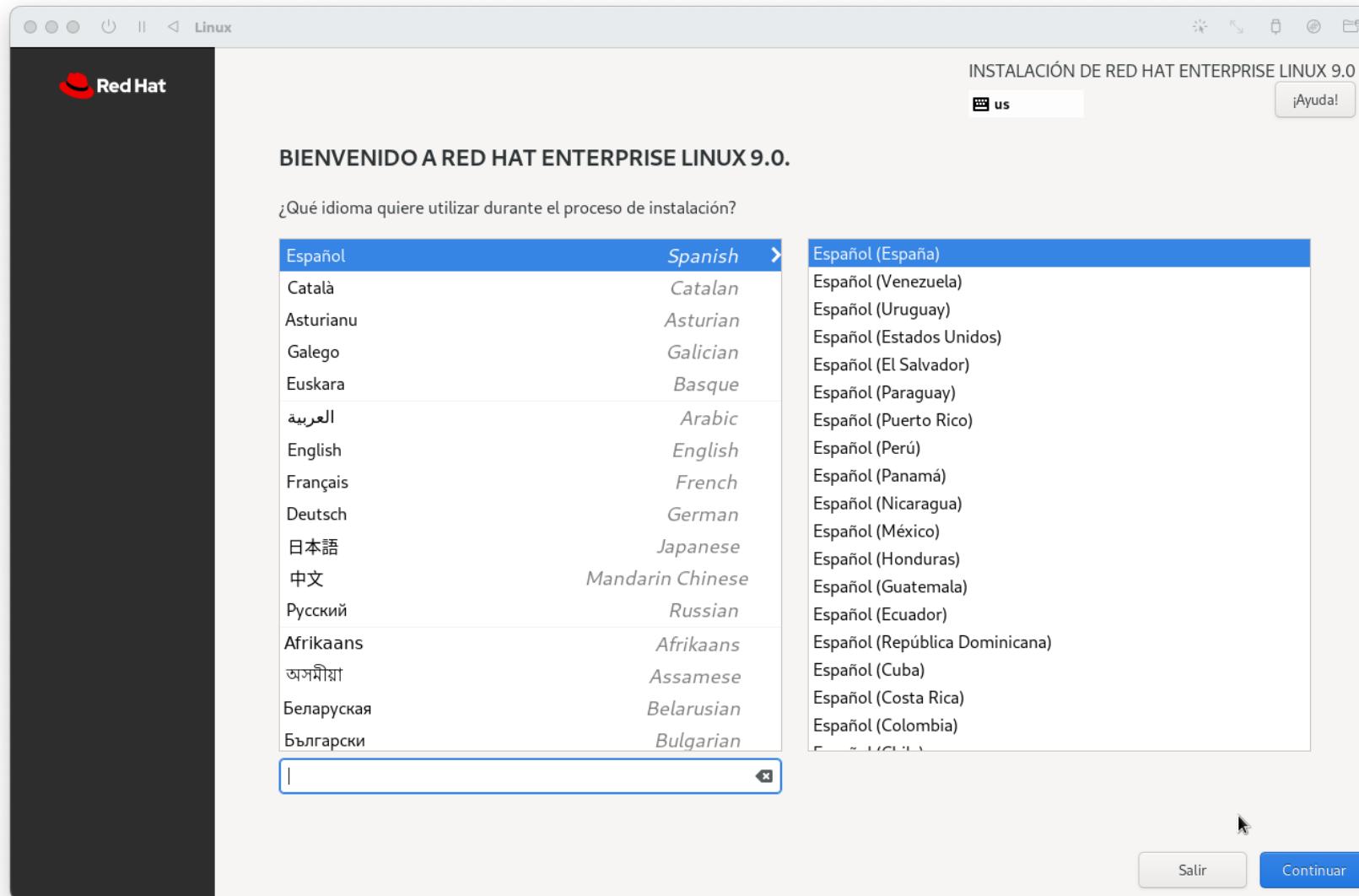
1. Instalación del SO
2. Post-instalación de SO
3. Tareas de administración básica
4. Configuración del arranque en servidores con BIOS/UEFI
5. Arranque desde medios externos
6. Bootloaders
7. Inicio y detención manual y automática de servicios

Práctica 1: Instalación de RHEL 9 / WS2019-22. Administración básica.
Configuración del arranque.

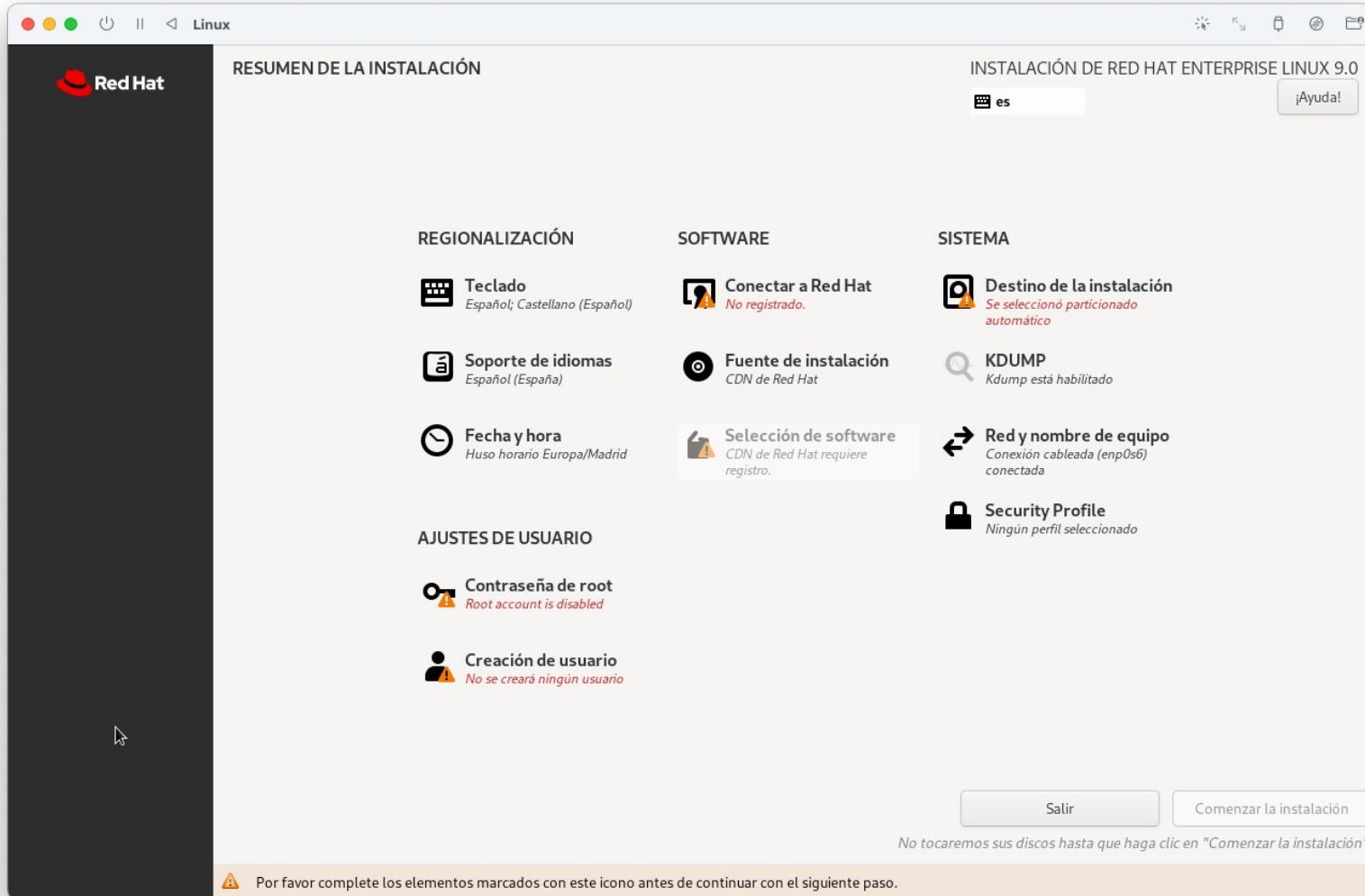
1.L Instalación de Linux (AlmaLinux/RHEL)

- En esta asignatura se emplea el **Red Hat Enterprise Linux**, derivado de Fedora. Utilizaremos una versión libre del mismo, **AlmaLinux** (<http://www.almalinux.org>)
- El instalador que se usa en estas distribuciones es **Anaconda**. Anaconda es un conjunto de módulos escritos en Python y ficheros adicionales escritos en C, junto con unidades de **systemd** y librerías de **dracut** (ambos se explican más adelante en este bloque).
- La instalación de AlmaLinux/RHEL puede ser **gráfica** o **automatizada**.
- La instalación gráfica puede ser **completa** o **mínima**. En la instalación completa se usa un USB/DVD que contiene todos los paquetes software. En la instalación mínima el medio de instalación contiene solamente los archivos necesarios para que el equipo se inicie y el programa Anaconda se ejecute. Los paquetes software pueden descargarse de un repositorio de internet (http o ftp), estar almacenados en una partición del disco o en un disco compartido en red.

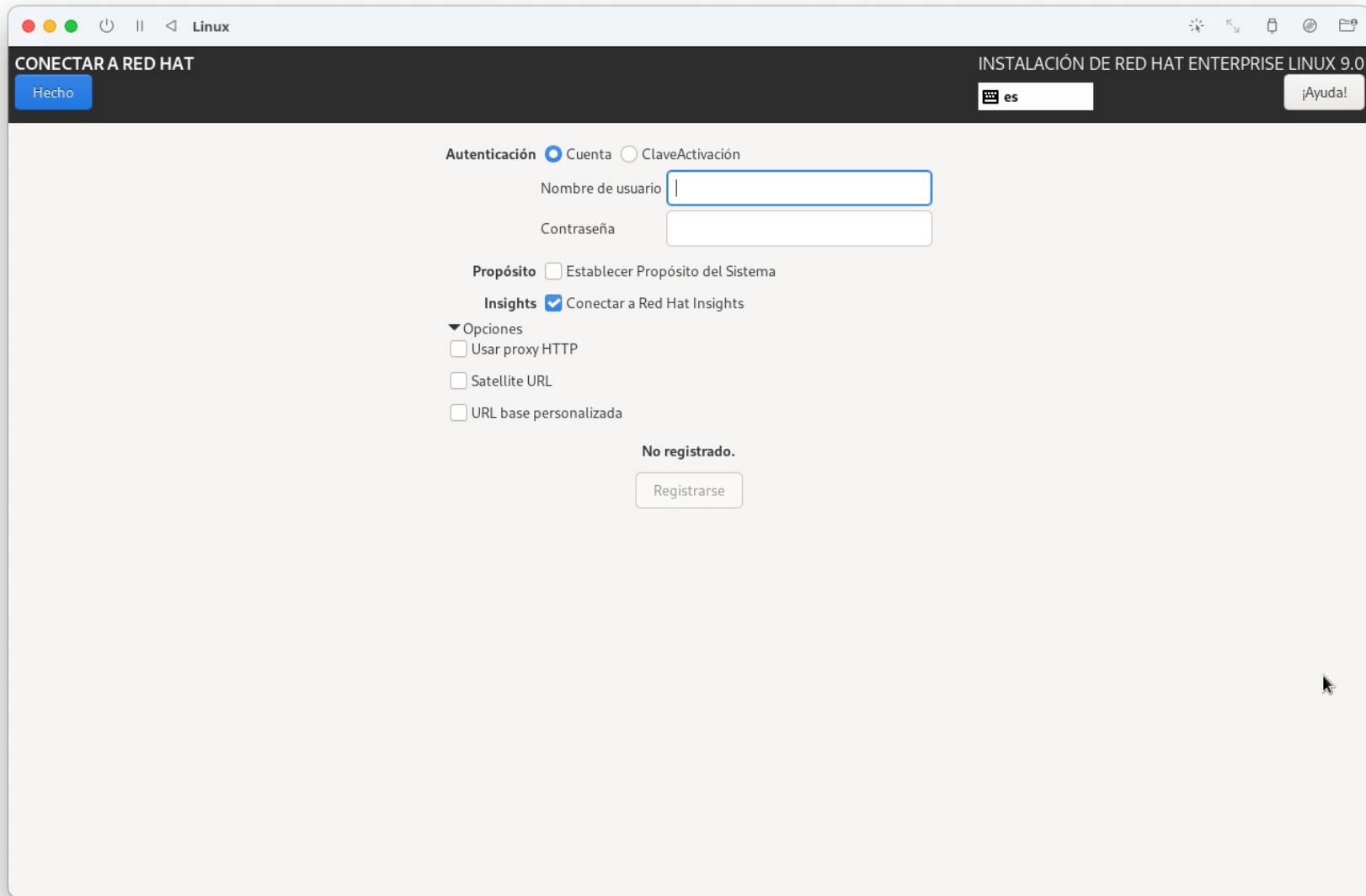
1.L Inicio de la instalación gráfica



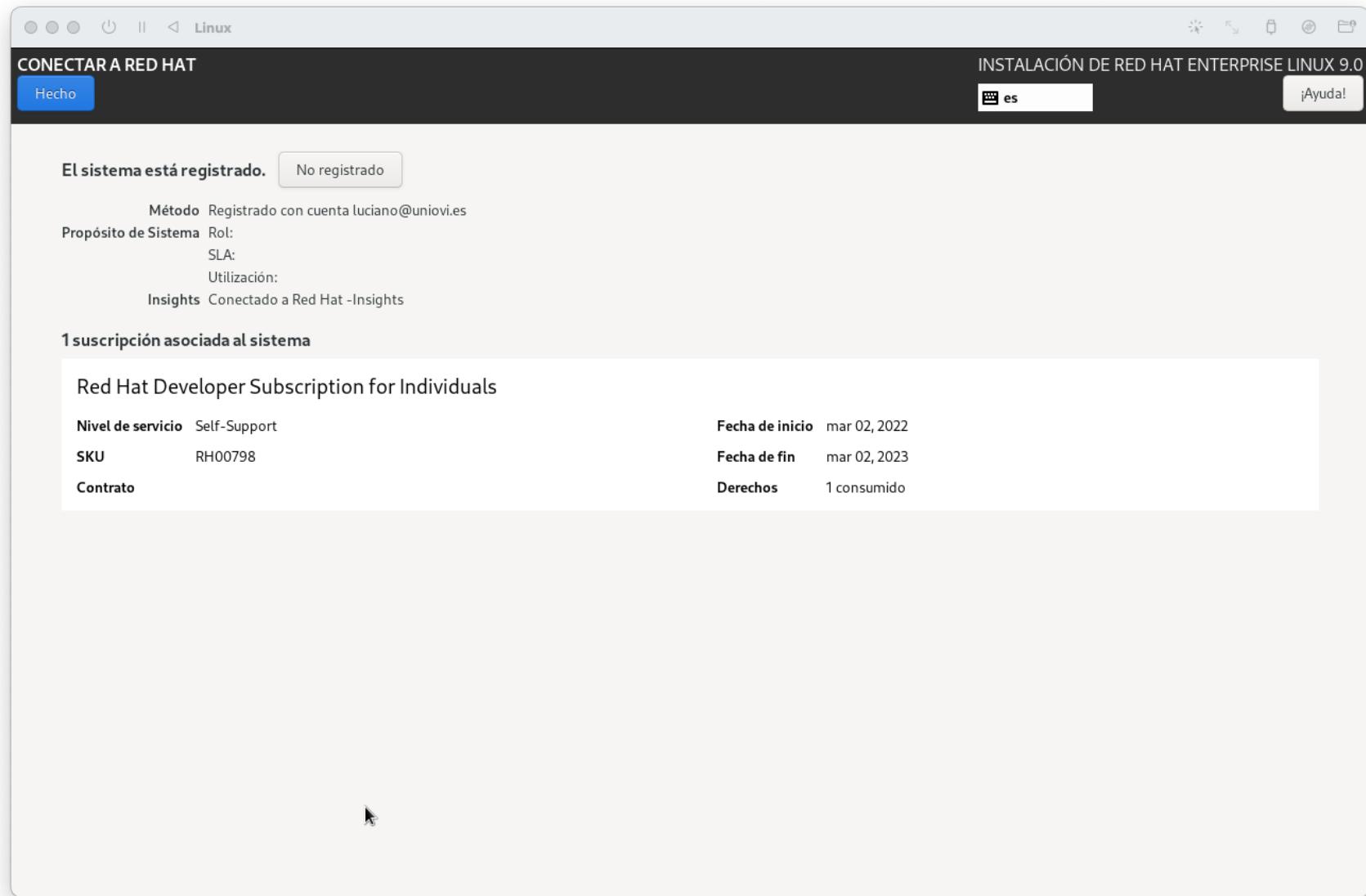
1.L Sumario de la instalación: RHEL 9



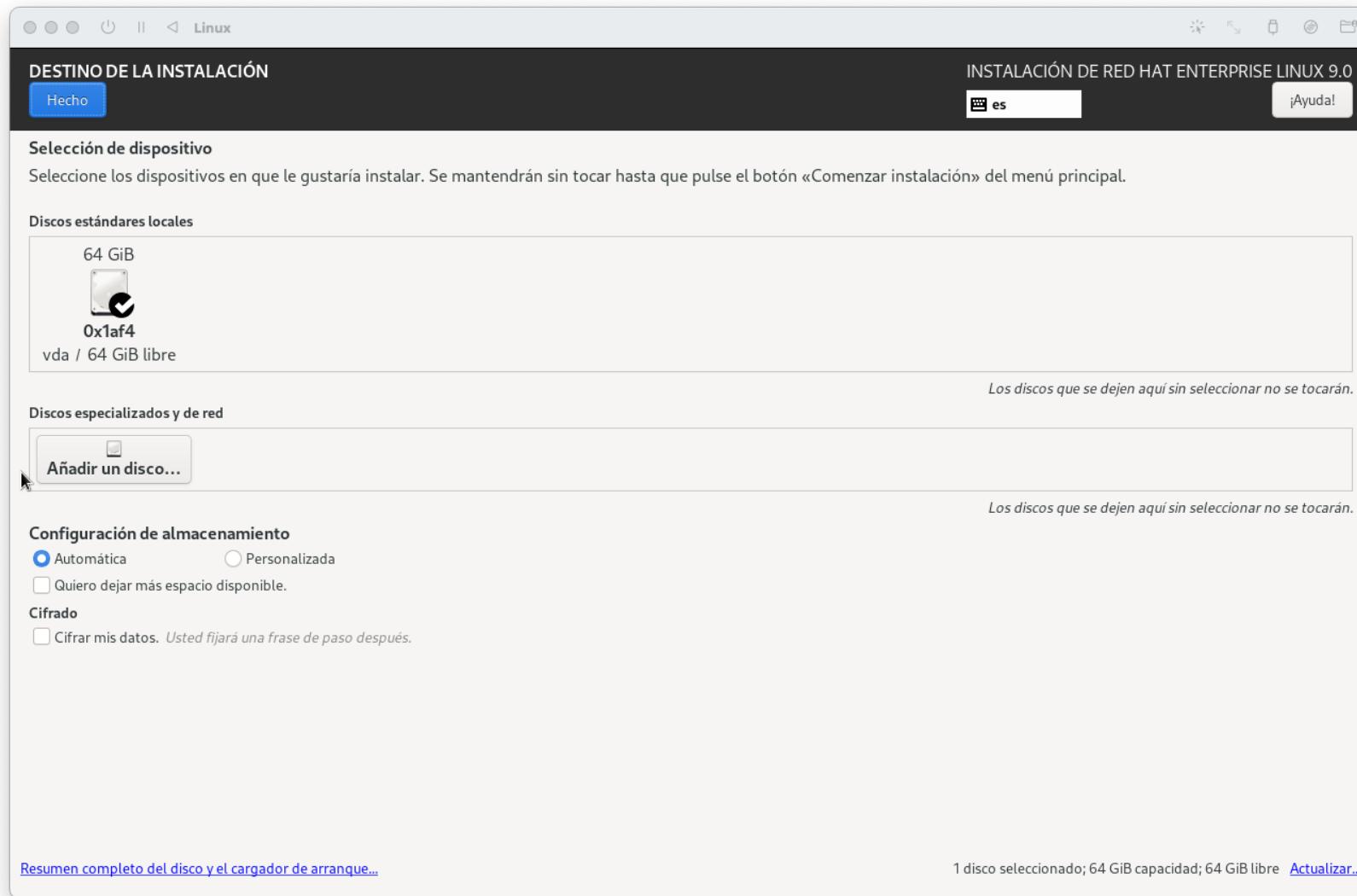
1.L Registrado de la instalación: RHEL 9



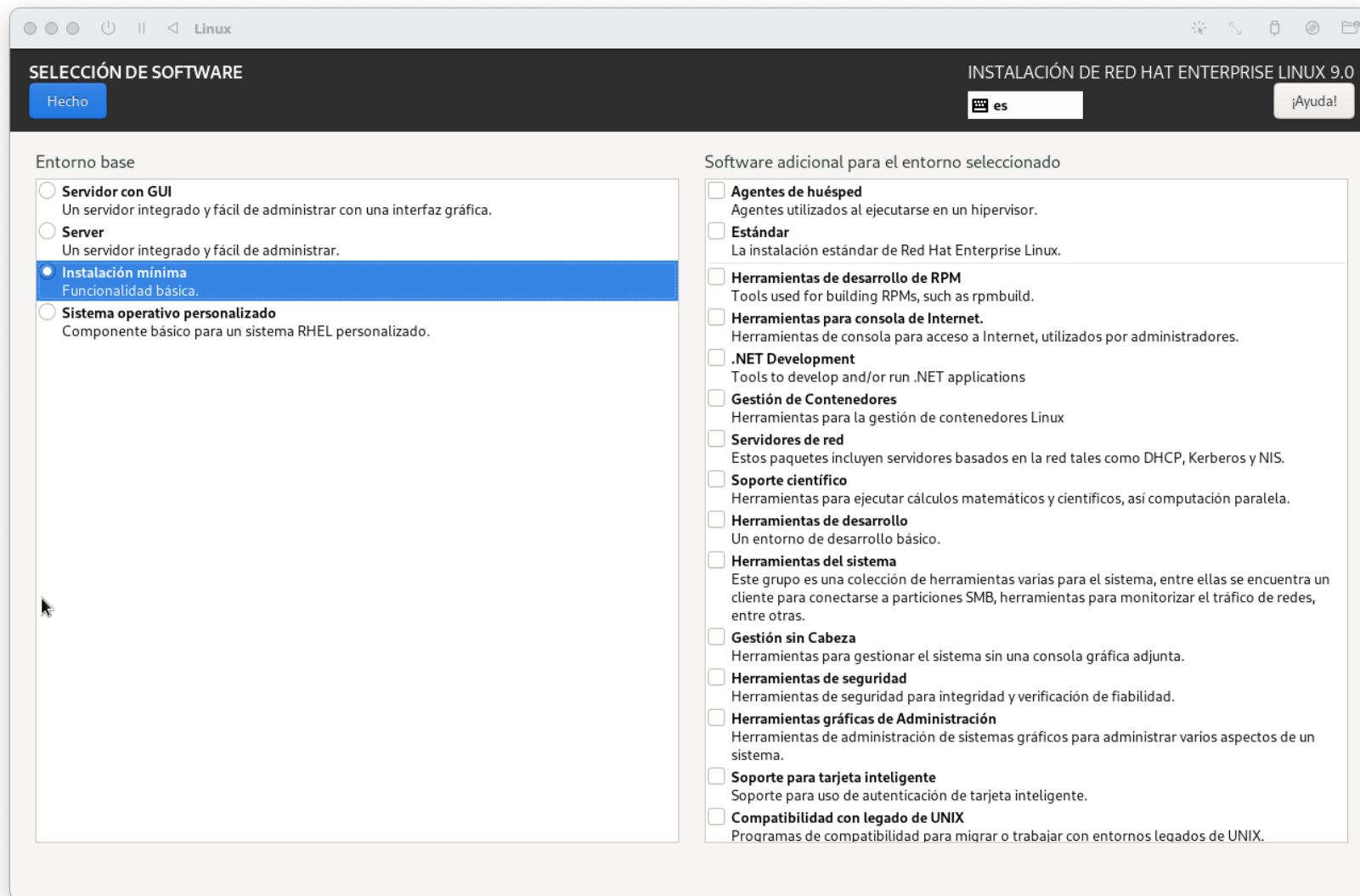
1.L Registrado de la instalación: RHEL 9



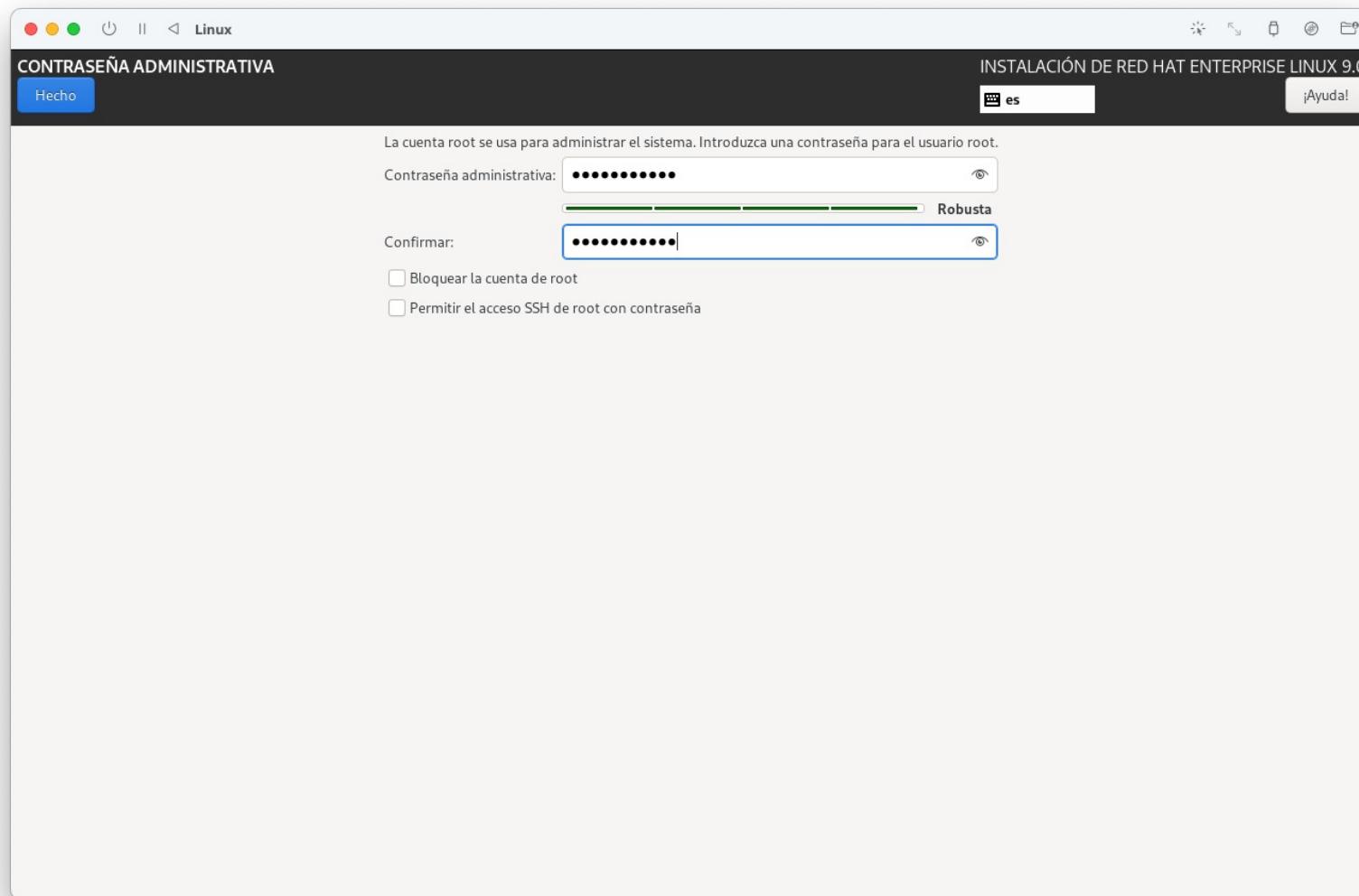
1.L Destino de la instalación: RHEL 9



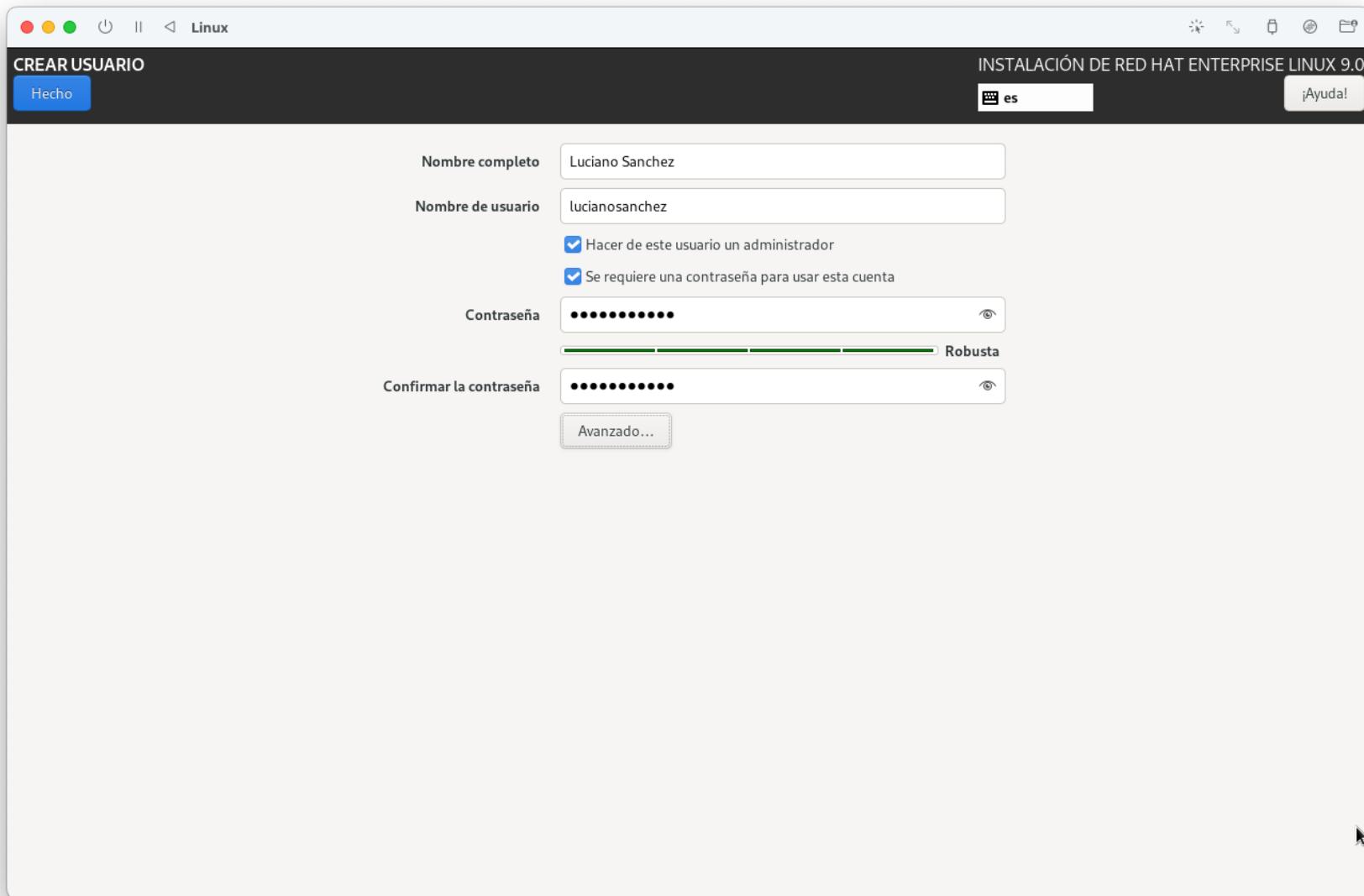
1.L Selección de software: RHEL 9



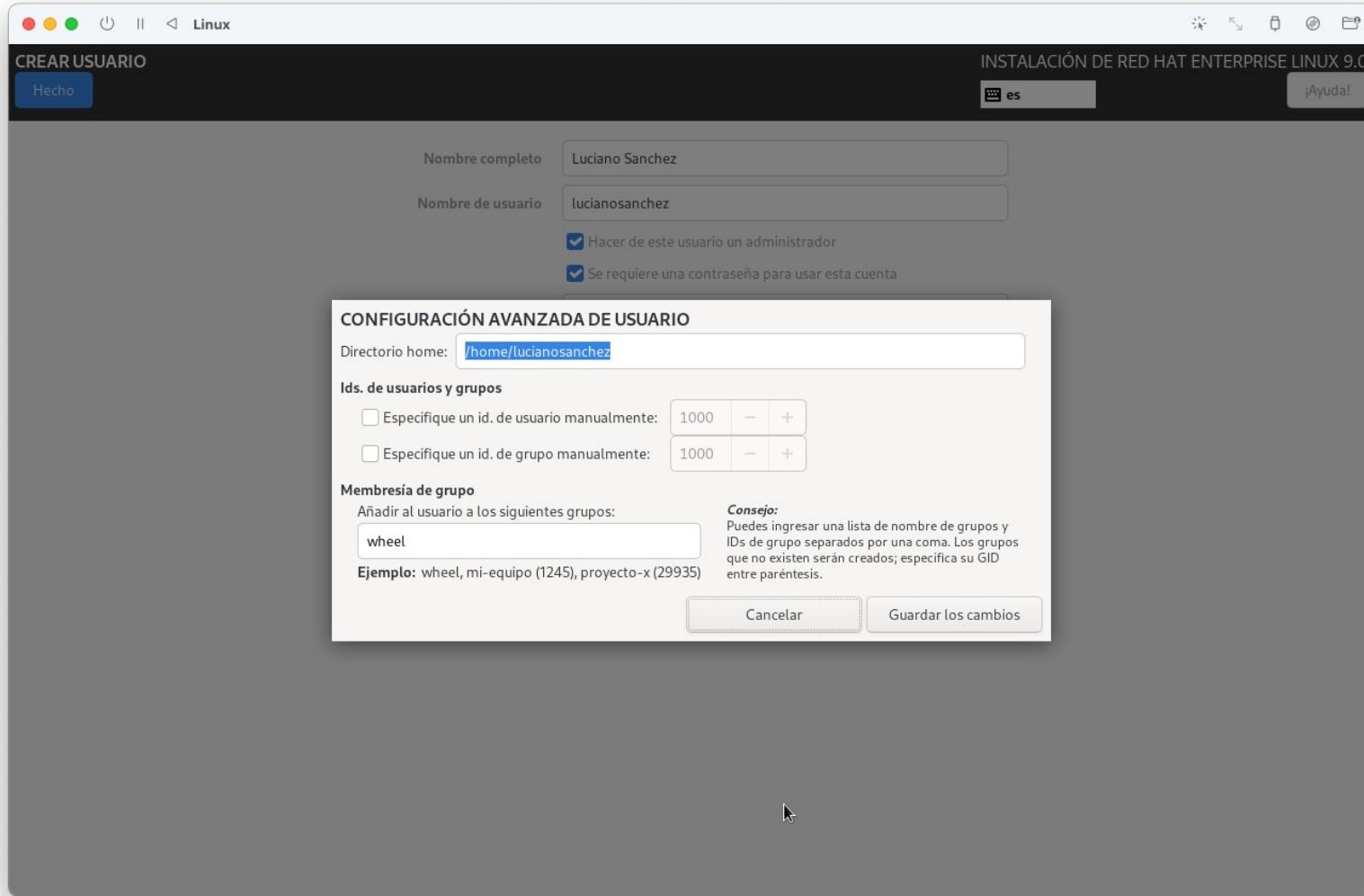
1.L Cuentas de administrador y usuario: RHEL 9



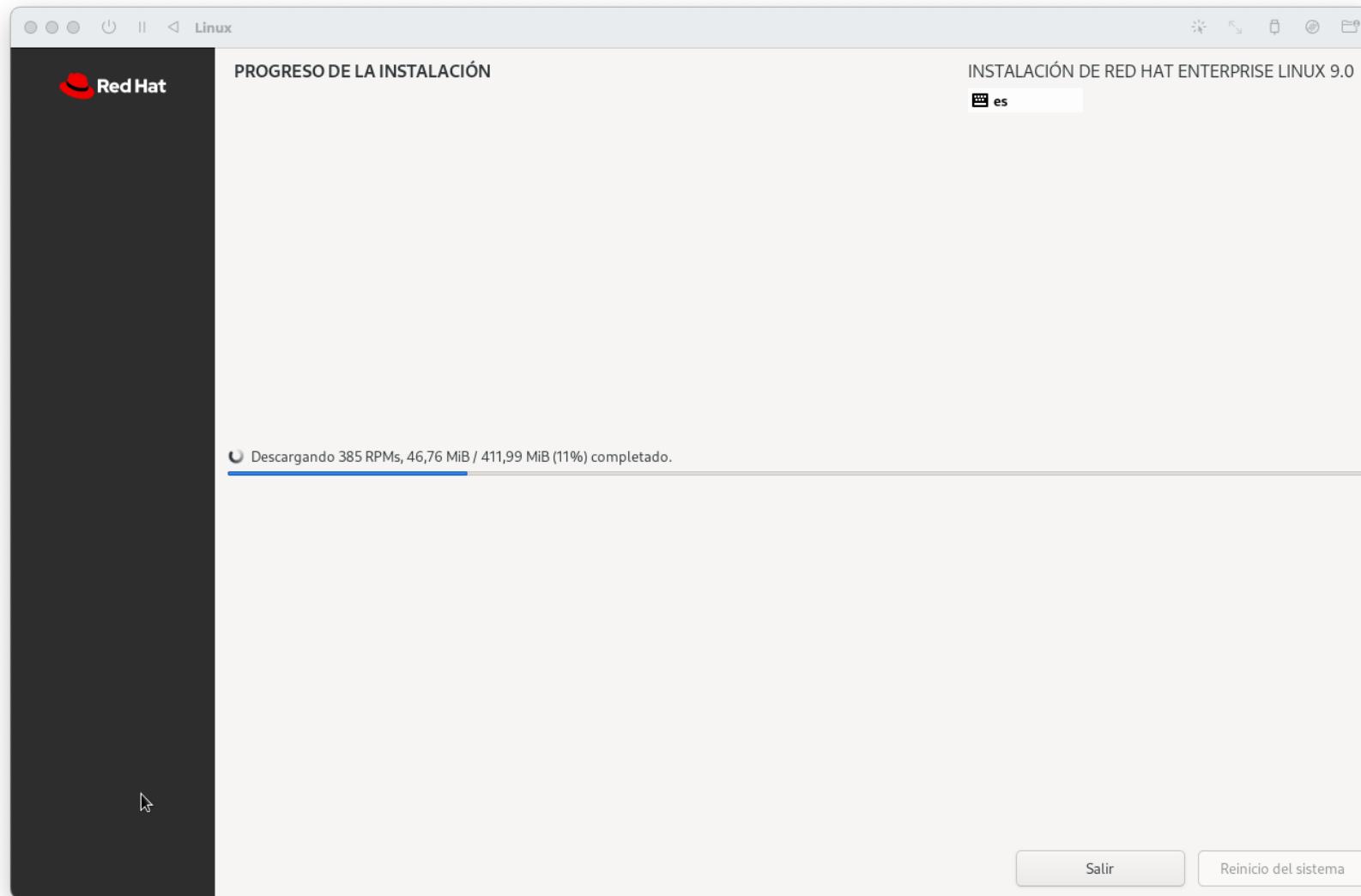
1.L Cuentas de administrador y usuario: RHEL 9



1.L Cuentas de administrador y usuario: RHEL 9



1.L Cuentas de administrador y usuario: RHEL 9



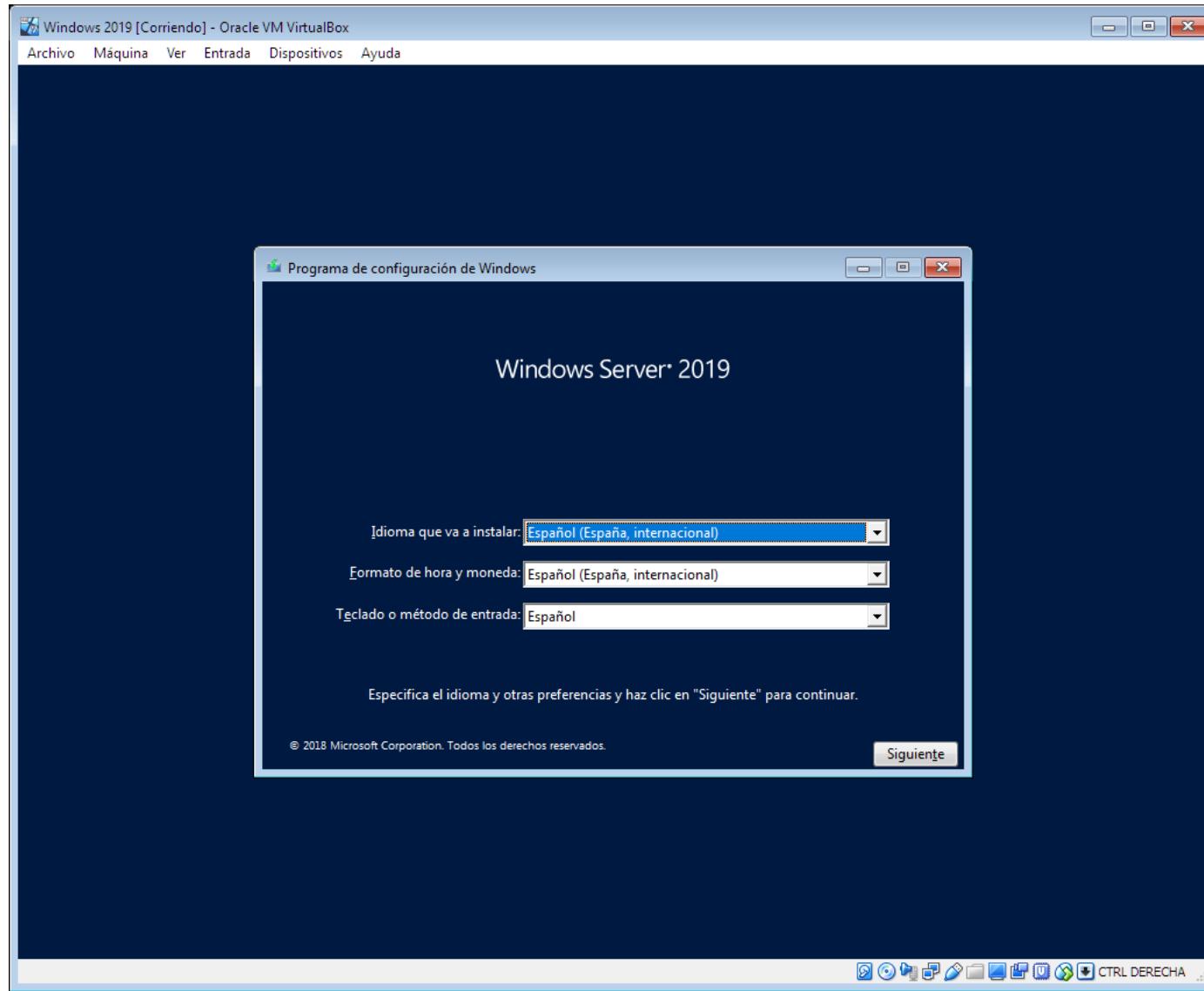
2.L Post-instalación de RHEL

- Los sistemas RHEL no son gratuitos. Se registran con la orden **subscription-manager** o con la herramienta gráfica correspondiente (Red Hat Subscription Manager). Se pueden ajustar parámetros de ejecución (servidor/workstation/nodo de computación), del nivel de servicio (Service Level Agreement, SLA) y de uso (producción / desarrollo / recuperación de desastres) con la orden **subscription-manager syspurpose**. Esta orden no es necesaria en AlmaLinux.
- En cuanto el sistema haya botado, por motivos de seguridad se deben actualizar todos los paquetes a su última versión con la orden **yum update**.
- El **firewall** debería estar instalado y activo por defecto. Se comprueba con **systemctl status firewalld**. Si no está activo, se puede forzar su arranque con las órdenes **systemctl start firewalld; systemctl enable firewalld**.
- Para mostrar los servicios activos, se emplea la orden **systemctl list-units | grep service**. Los servicios innecesarios pueden deshabilitarse con la orden **systemctl mask**.
- **SELinux** es una capa adicional de seguridad que determina qué procesos pueden acceder a los ficheros, directorios y puertos. Cuando SELinux está activado, puede estar en modo restrictivo (enforcing) o permisivo (permissive). El modo por defecto es el más seguro. Puede comprobarse el estado mediante la orden **getenforce**.

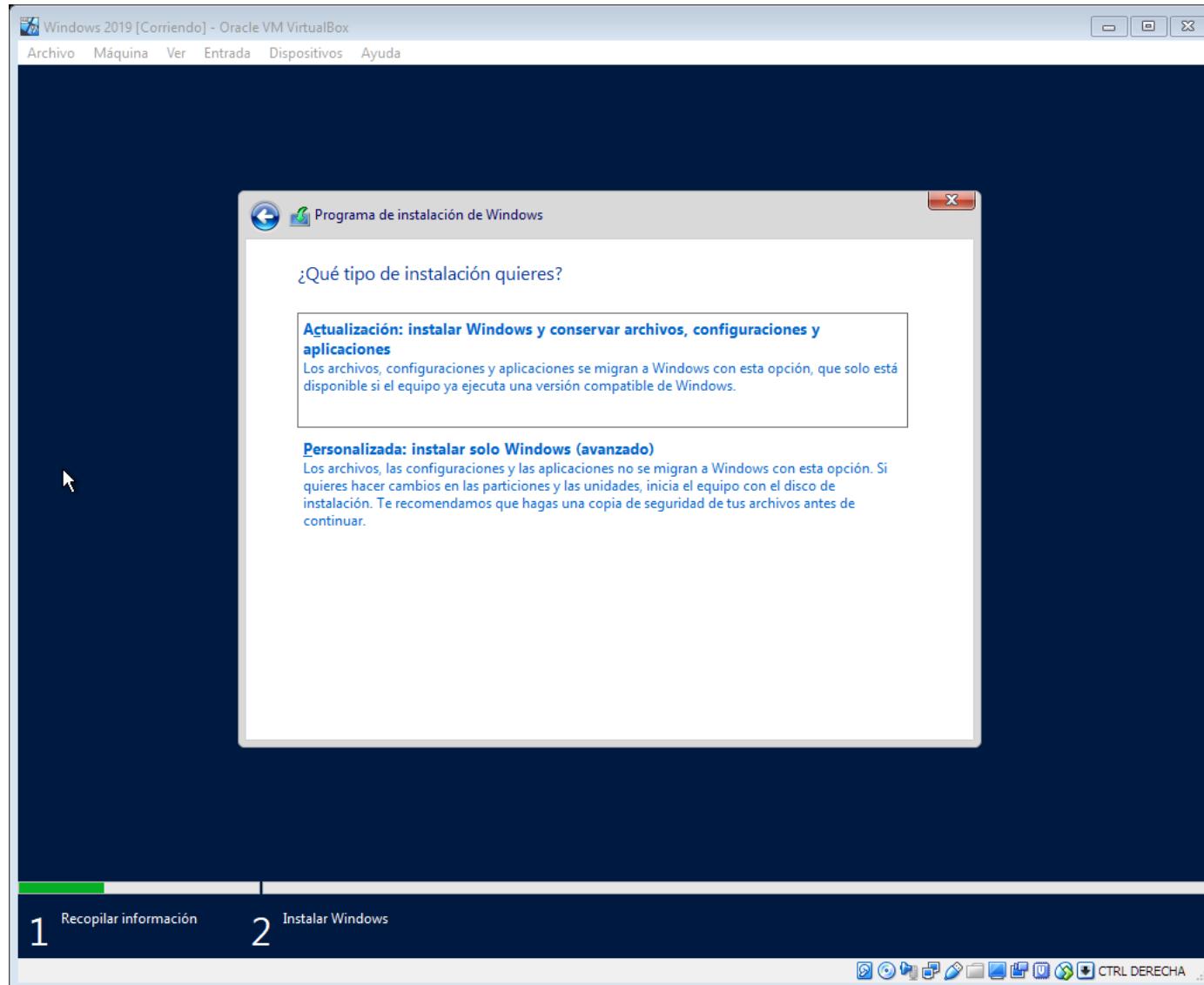
1.W Instalación de Windows 2019

- En esta asignatura se emplean **Windows Server 2019/2022**
- La instalación de WS no es sustancialmente diferente de la instalación de Windows 10/11
- Puede decidirse entre un sistema completo o una versión mínima, sólo en modo texto (**core server**). La instalación y la administración de un servidor windows, al igual que ocurría con los servidores Linux, raramente se realiza mediante un teclado y una pantalla conectados al equipo. En la instalación se pueden emplear tarjetas administrativas y una vez instalado la conexión puede realizarse con **Remote Desktop** o mediante un navegador web si se usa **Windows Admin Center**.

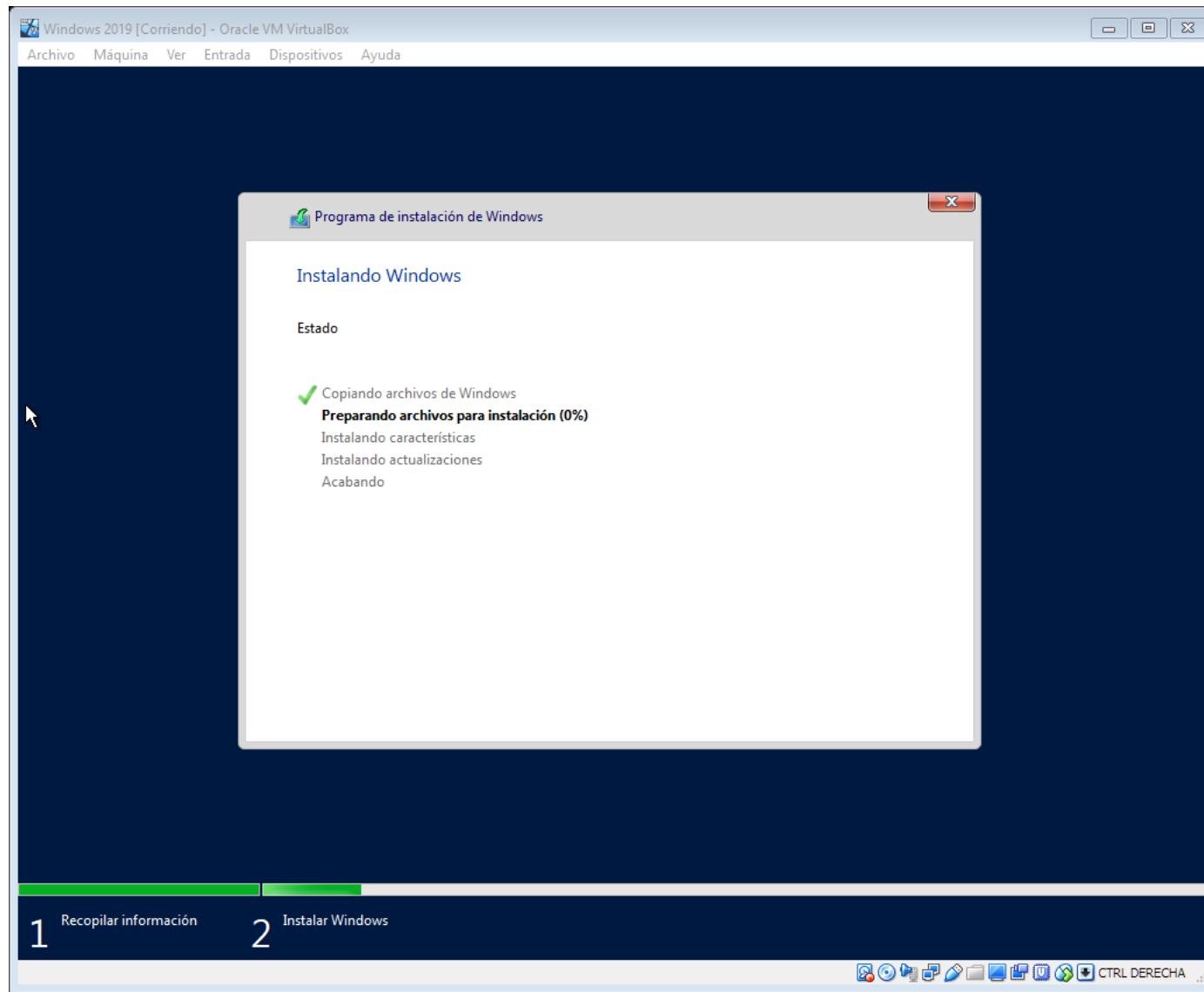
1.W Instalación de Windows 2019



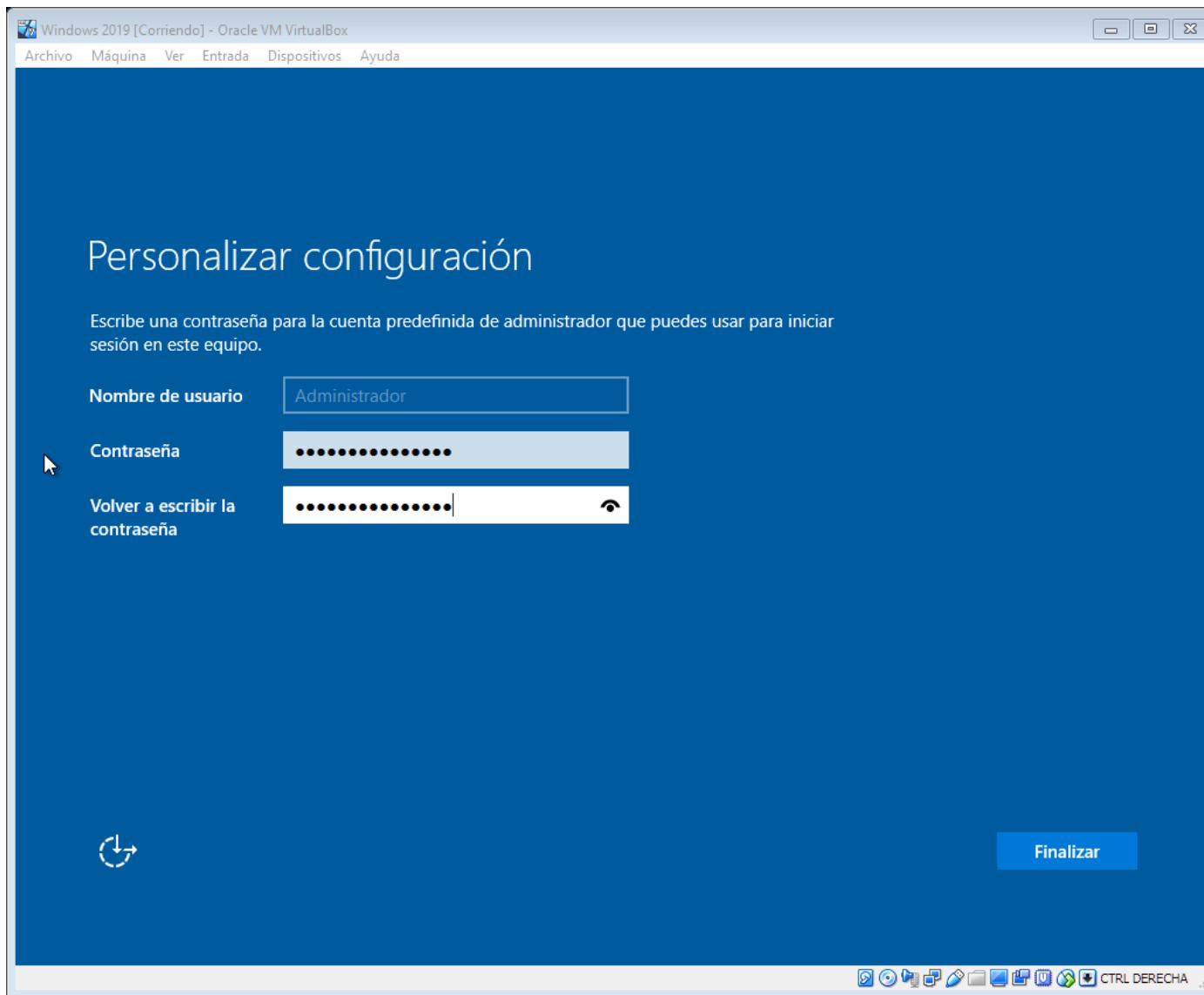
1.W Selección del tipo de instalación



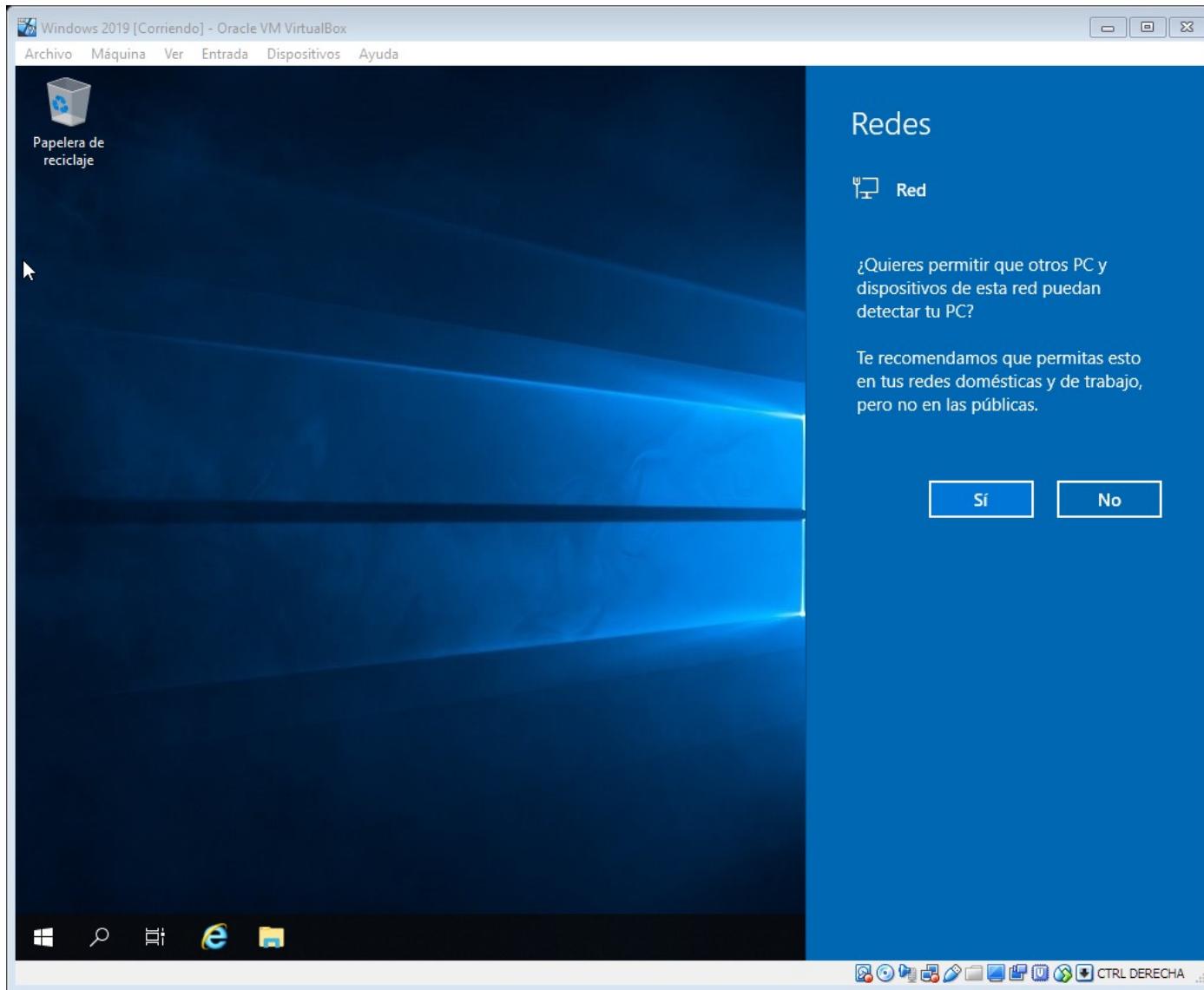
1.W Instalando Windows



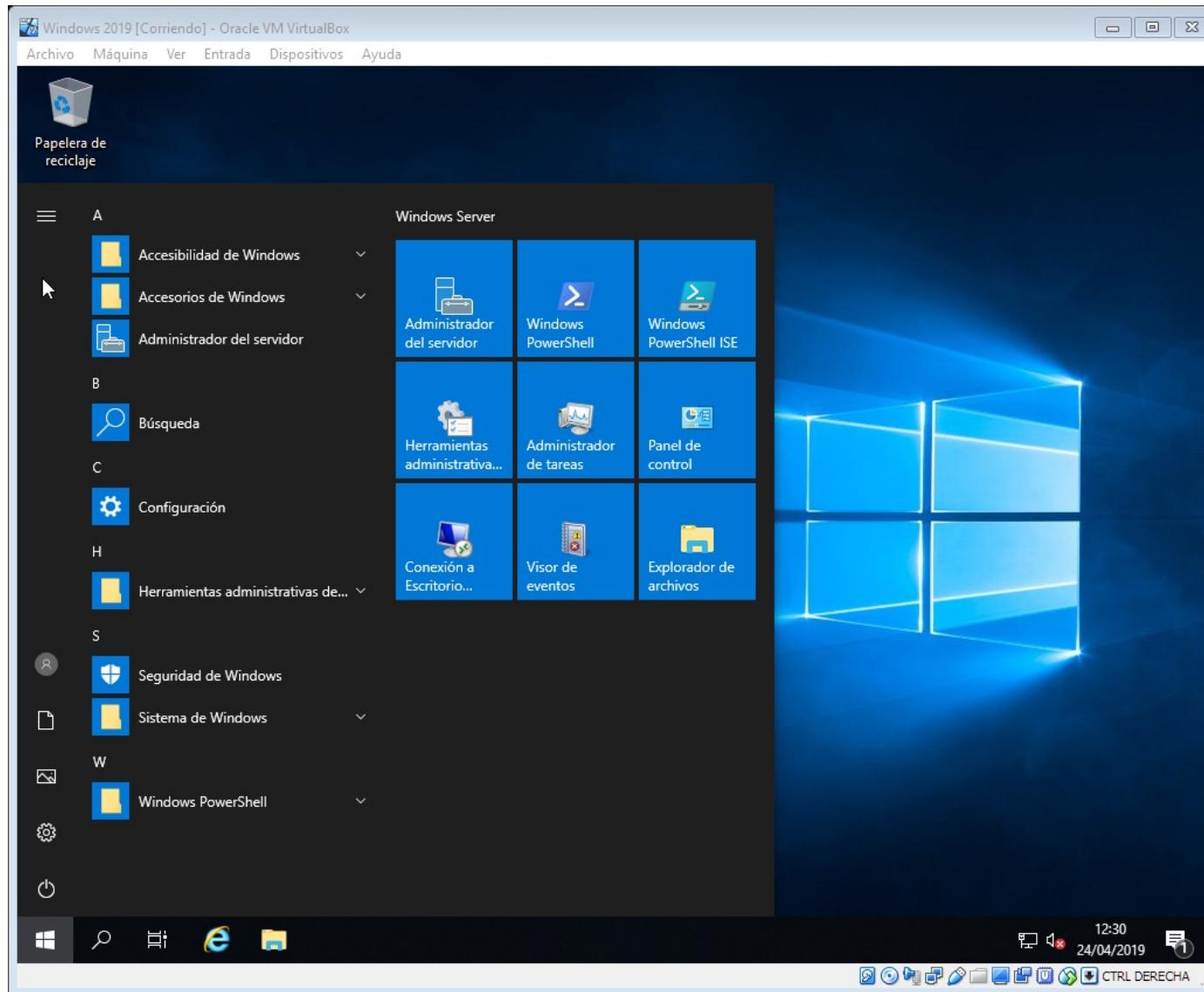
1.W Instalando Windows



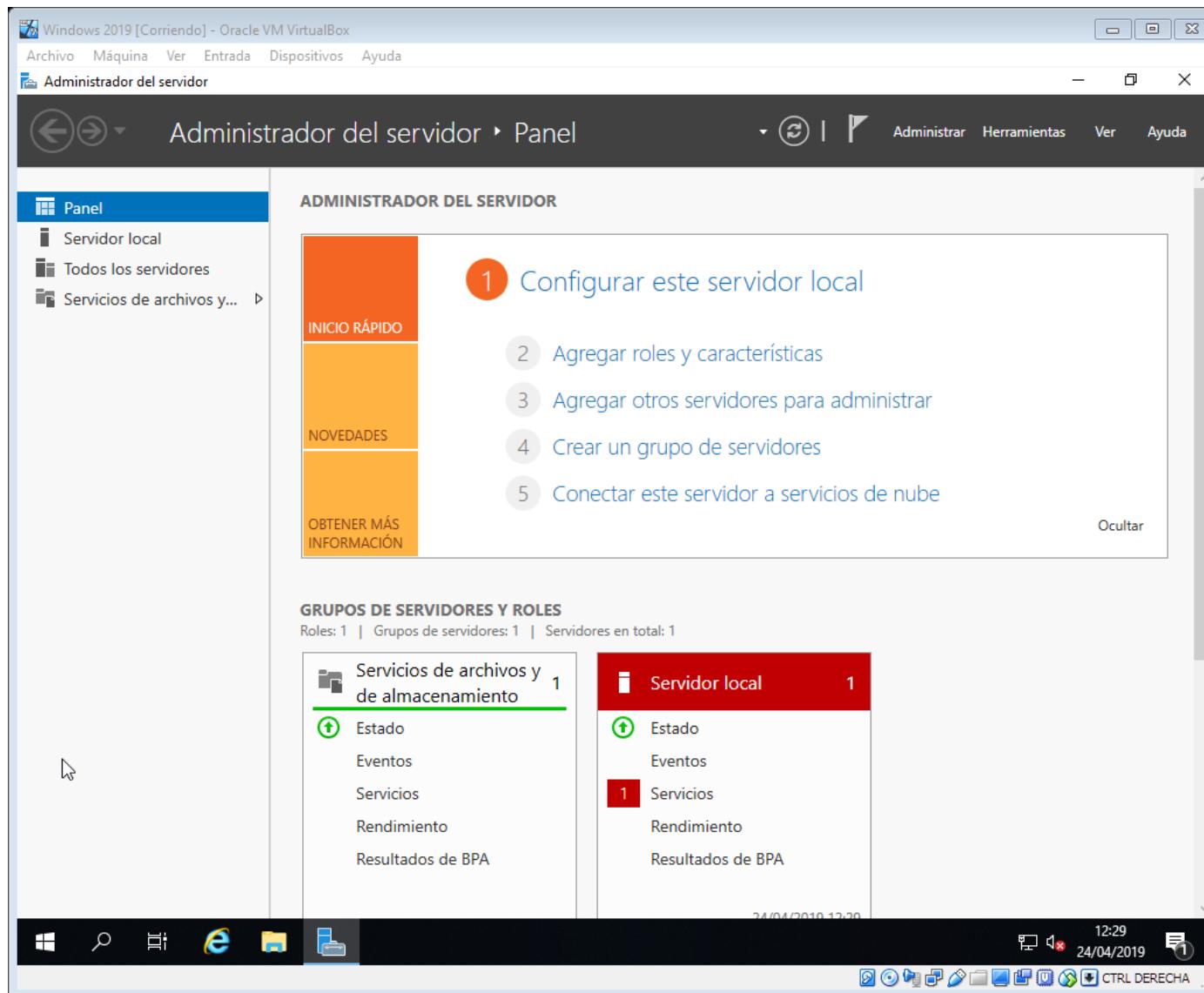
1.W Instalando Windows



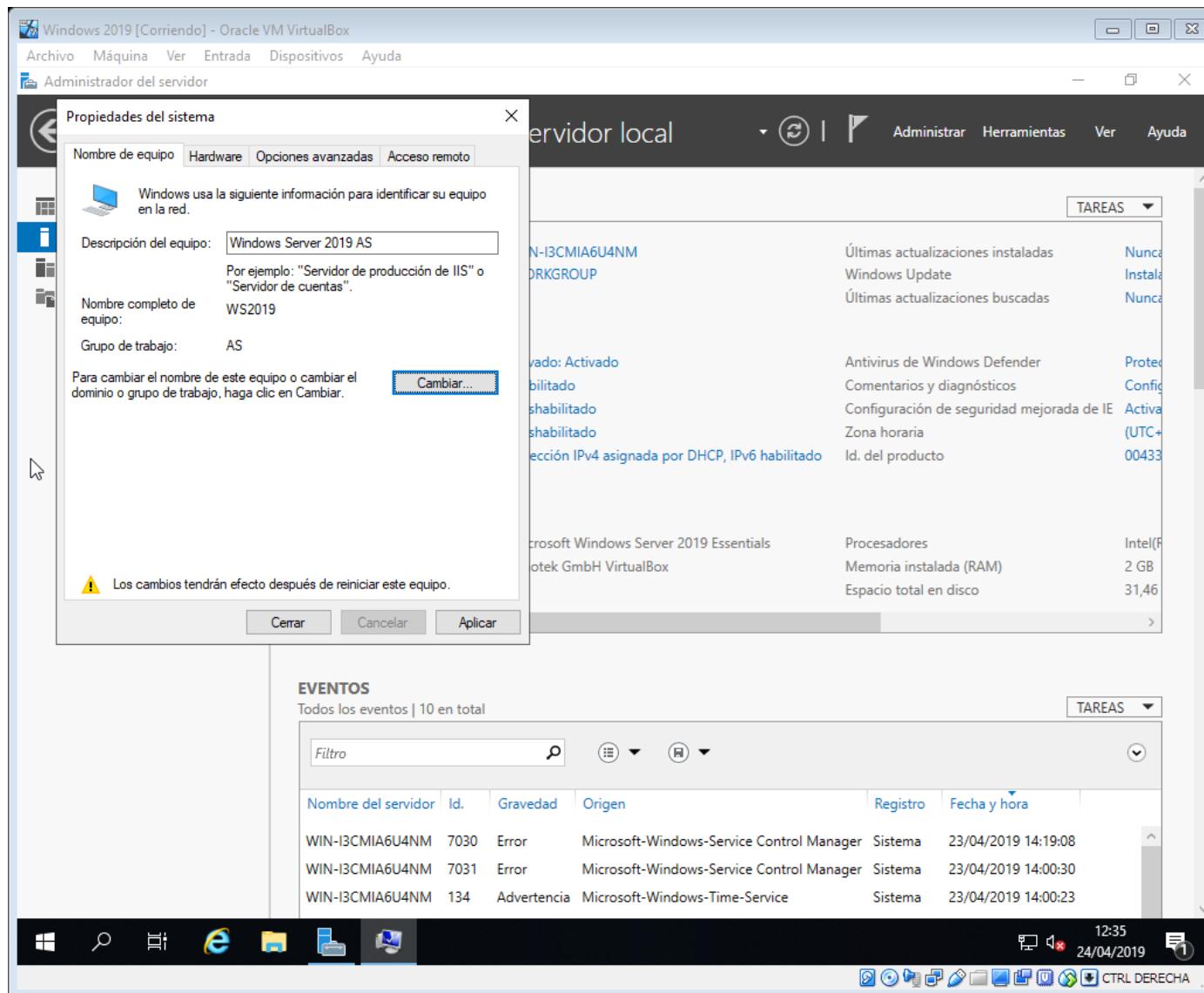
2.W Post-instalación: Administrador del servidor



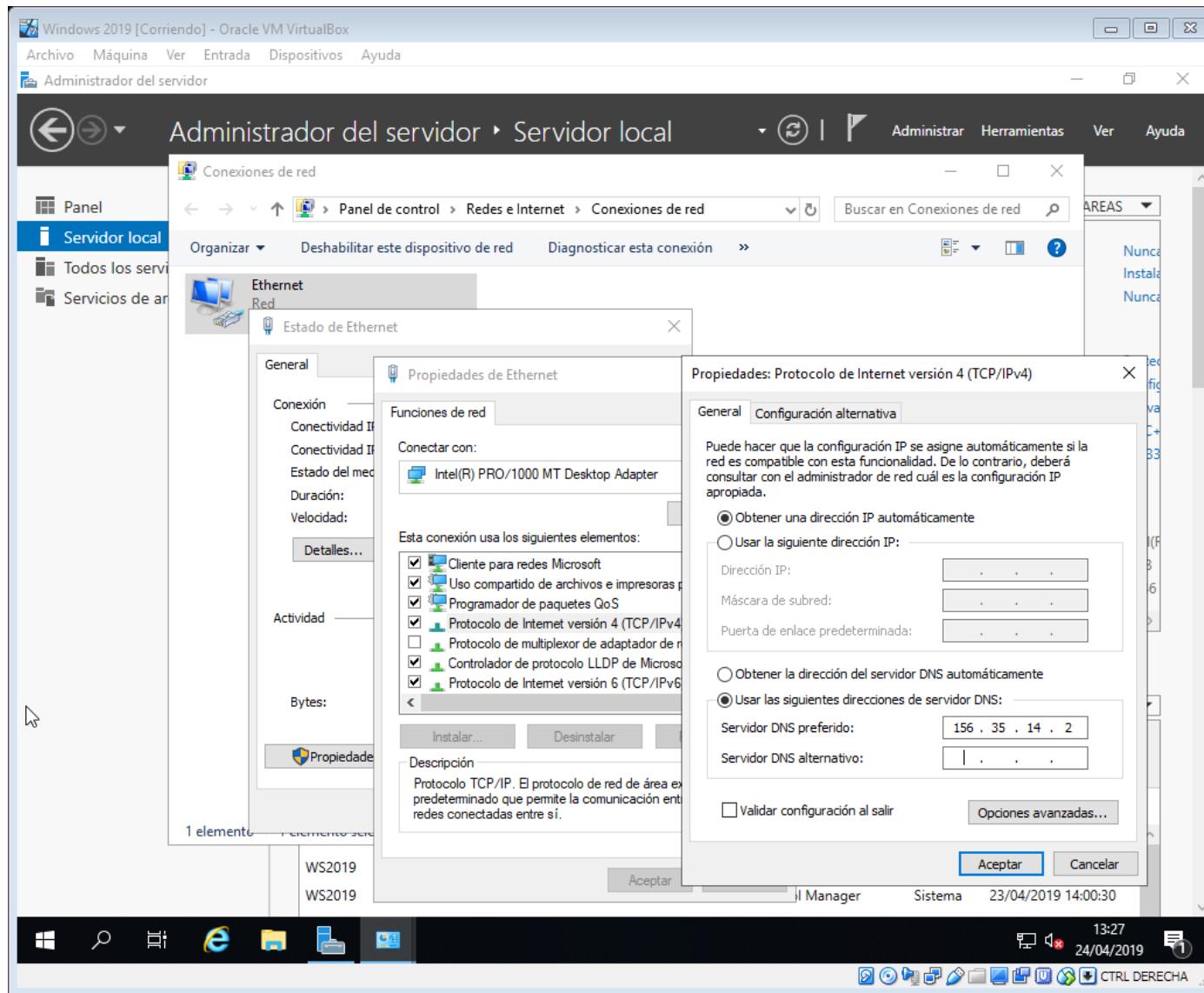
1.W Post-instalación: Administrador del servidor



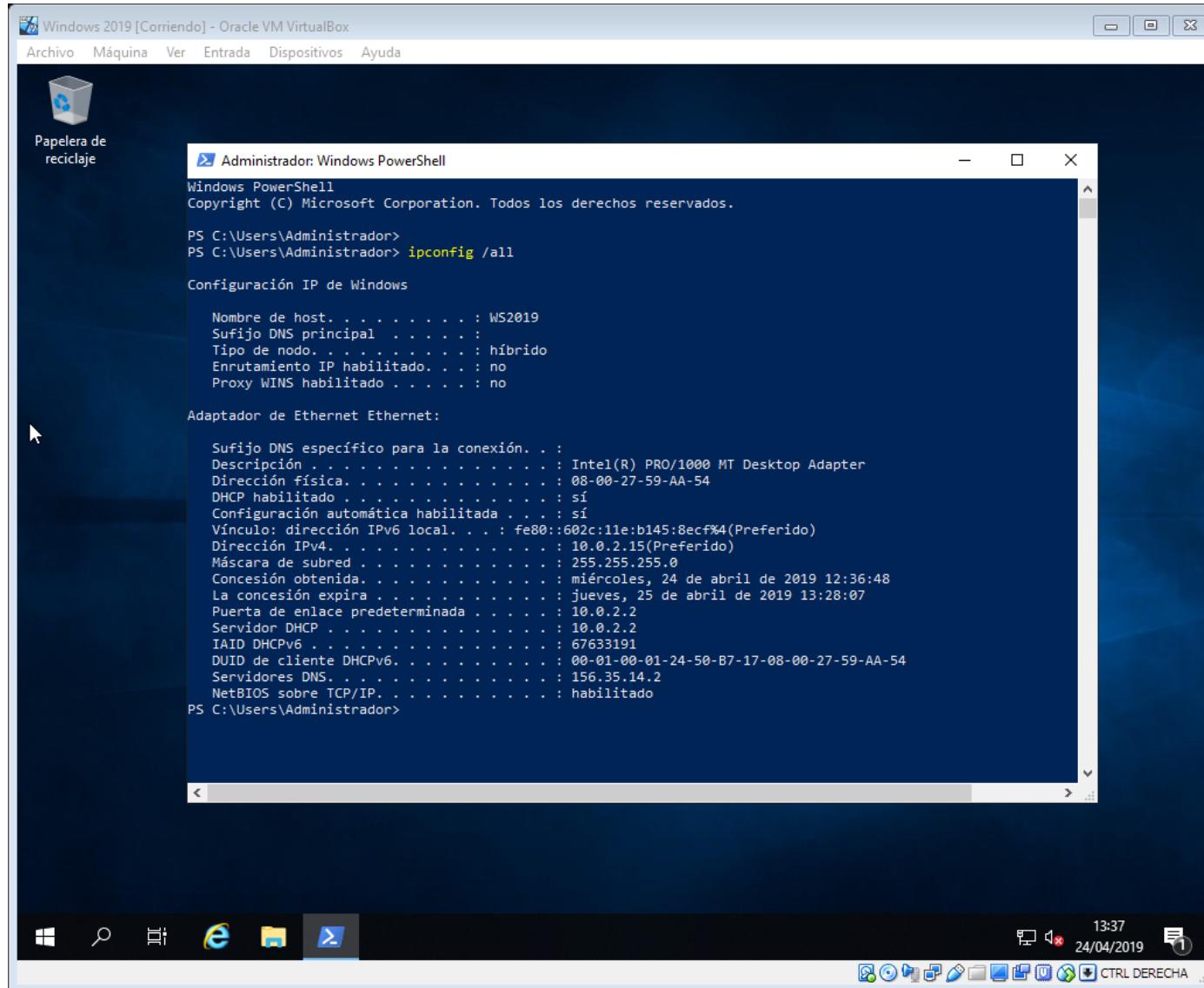
2.W Post-instalación: Servidor local -> configurar



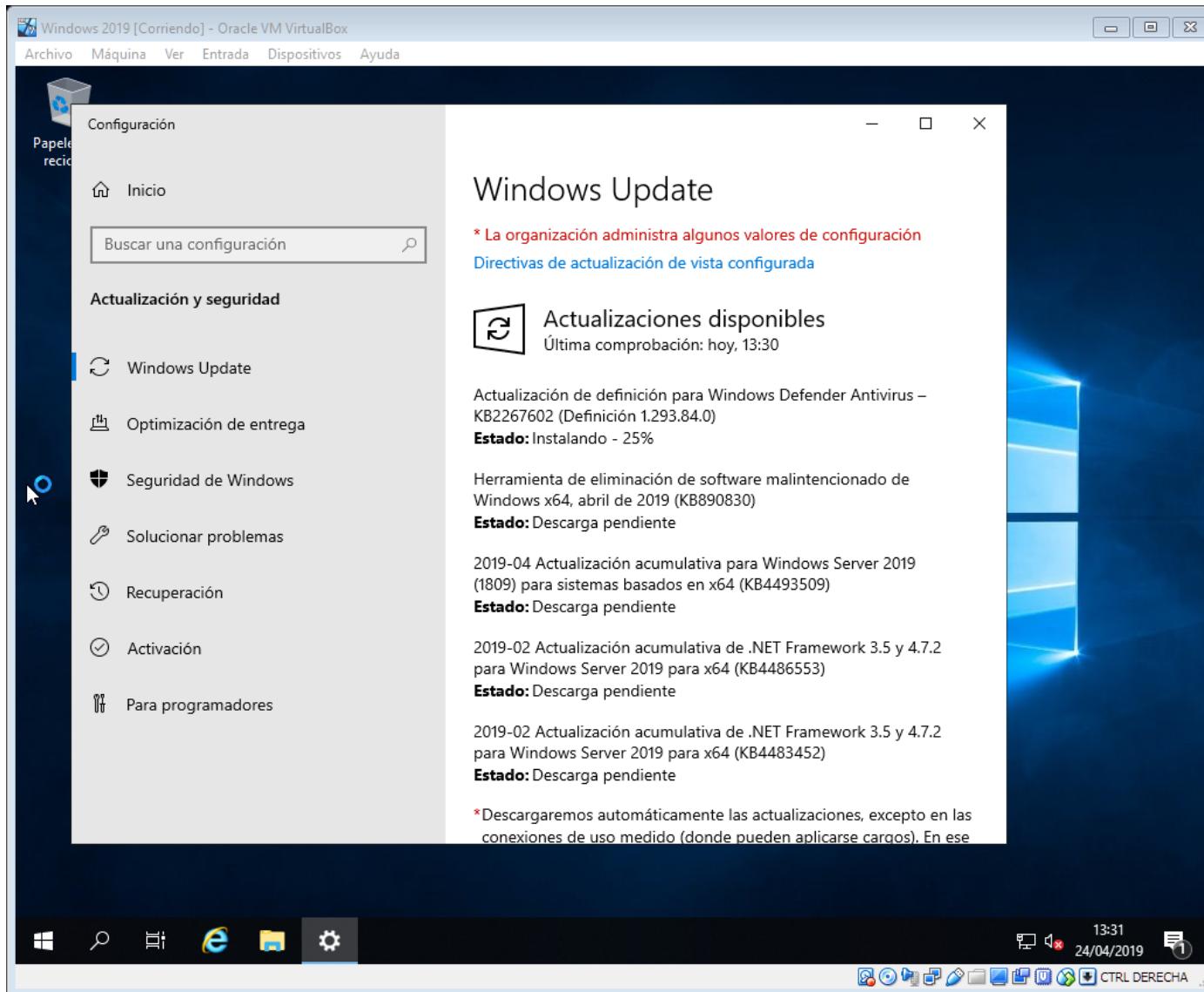
2.W Post-instalación: Servidor local -> configurar red



2.W Post-instalación: Comprobación configuración red



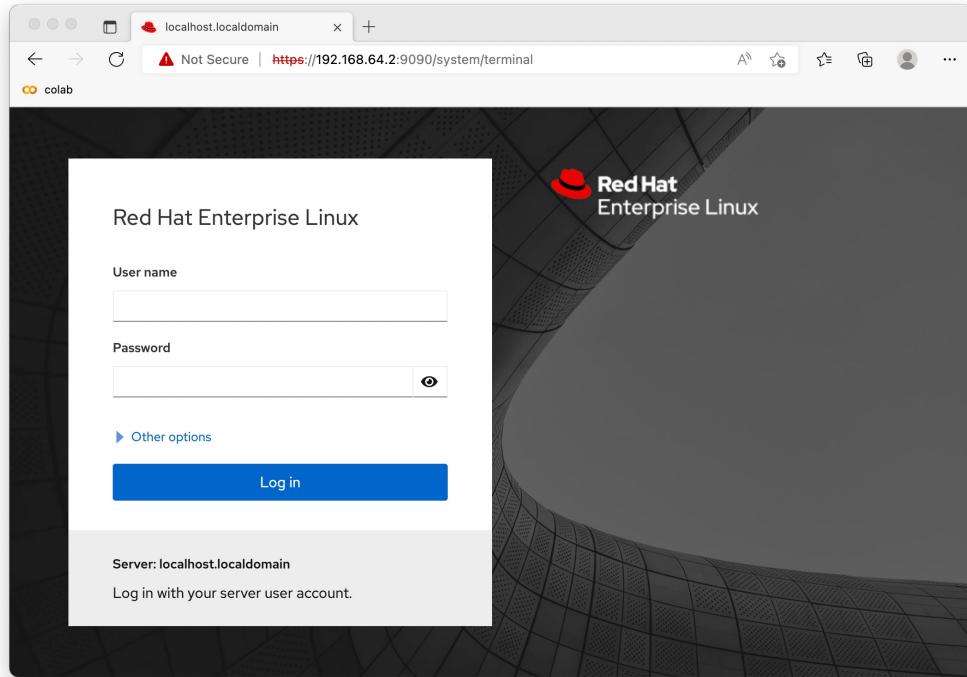
2.W Post-instalación: Actualización del sistema



3.L Tareas de administración básica RHEL

- RHEL puede administrarse desde una consola de texto o desde una consola web. También es posible administrar el sistema desde una consola gráfica si el equipo es una workstation o dispone de una tarjeta de administración remota.
- La mayoría de las tareas pueden hacerse desde la consola web. Entre otras, pueden consultarse las características del sistema (hardware, perfiles de funcionamiento, conexiones), los mensajes del sistema, los interfaces de red y el firewall, instalar paquetes, actualizar el sistema, configurar el nivel de seguridad con SELinux y acceder al terminal.
- Si la consola no está instalada por defecto, puede instalarse con **yum install cockpit**
- Si no está iniciada automáticamente, se inicia con **systemctl enable --now cockpit.socket**
- El servidor escucha en el puerto 9090. Si el servidor está virtualizado, para poder acceder desde el exterior al interfaz de red debe estar en modo **bridged** (o **host-only** si se configura adecuadamente la red en la máquina virtual)

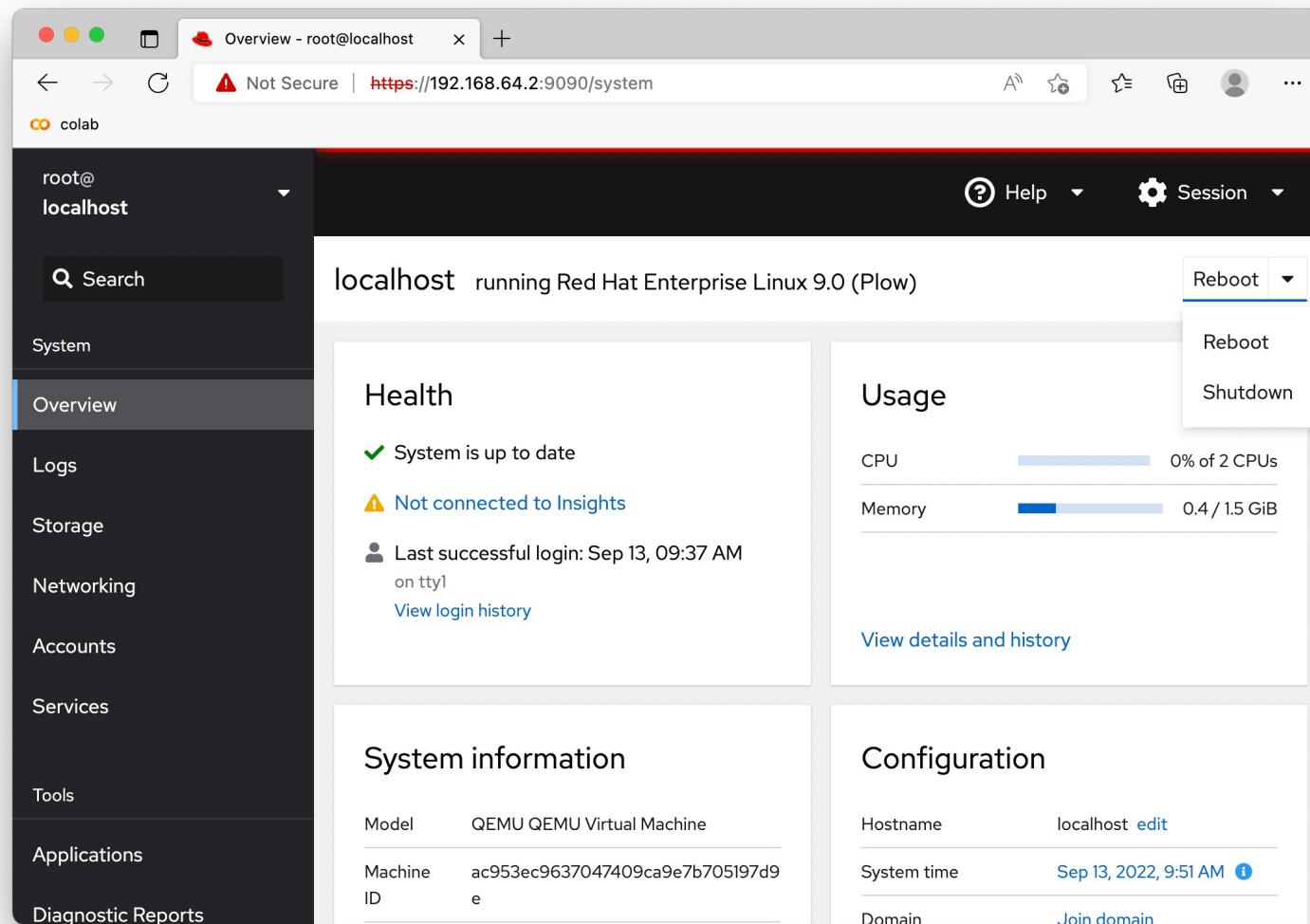
3.L Tareas de administración básica RHEL



- Si el firewall no tiene la configuración estándar, se abre el puerto 9090 con las ordenes **firewall-cmd --add-service=cockpit --permanent; firewall-cmd --reload**
- Si el usuario con que se hace login en la consola tiene privilegios de administración, hay que marcar la casilla “Reuse my password for privileged tasks” para que pueda instalarse software o cambiar la configuración de SELinux.

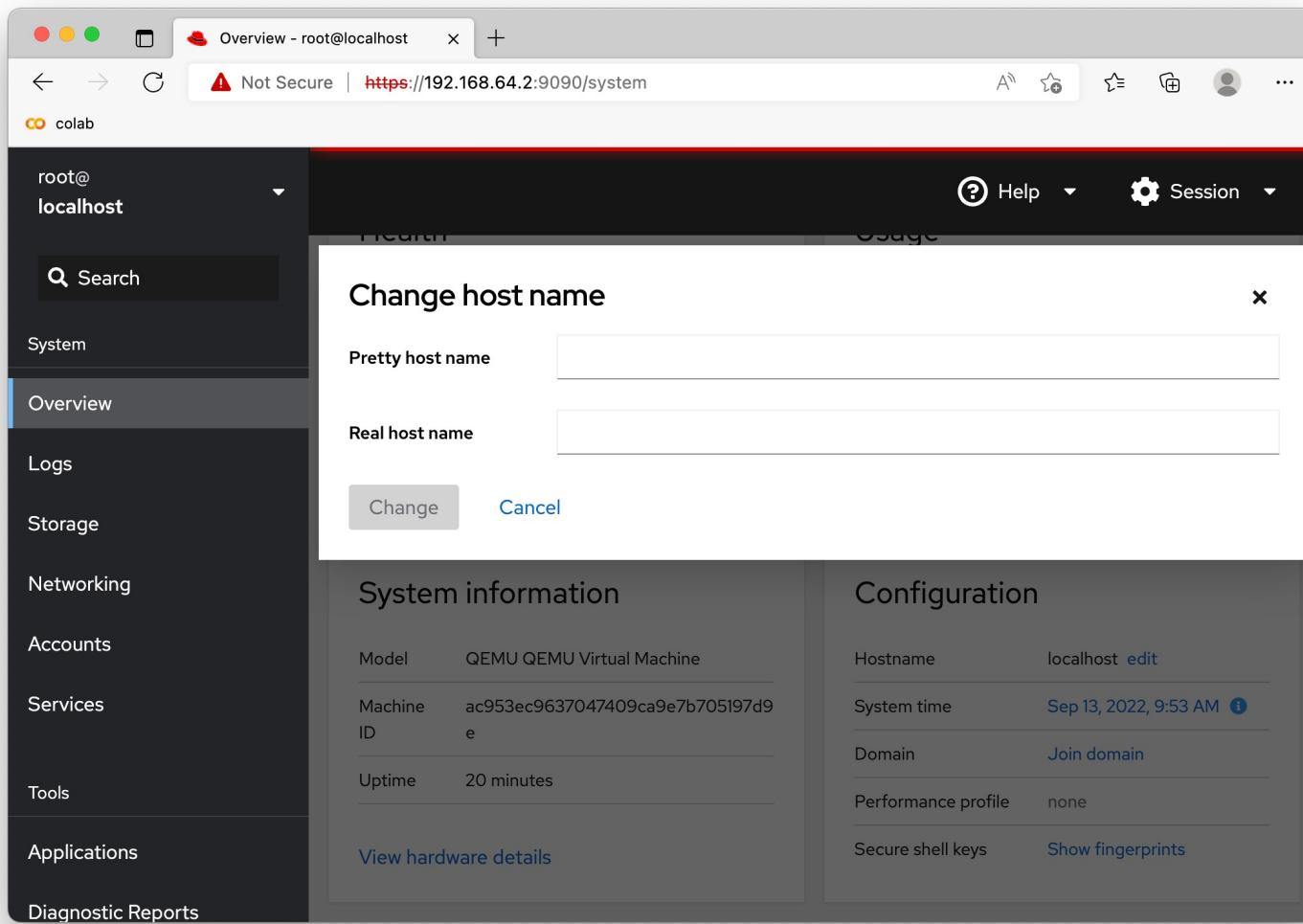
3.L Apagar y reiniciar el sistema

- Desde la consola web: en el menú Overview: Reboot / Shut down
- Desde el terminal: **systemctl poweroff** para apagar, **systemctl reboot** para reiniciar



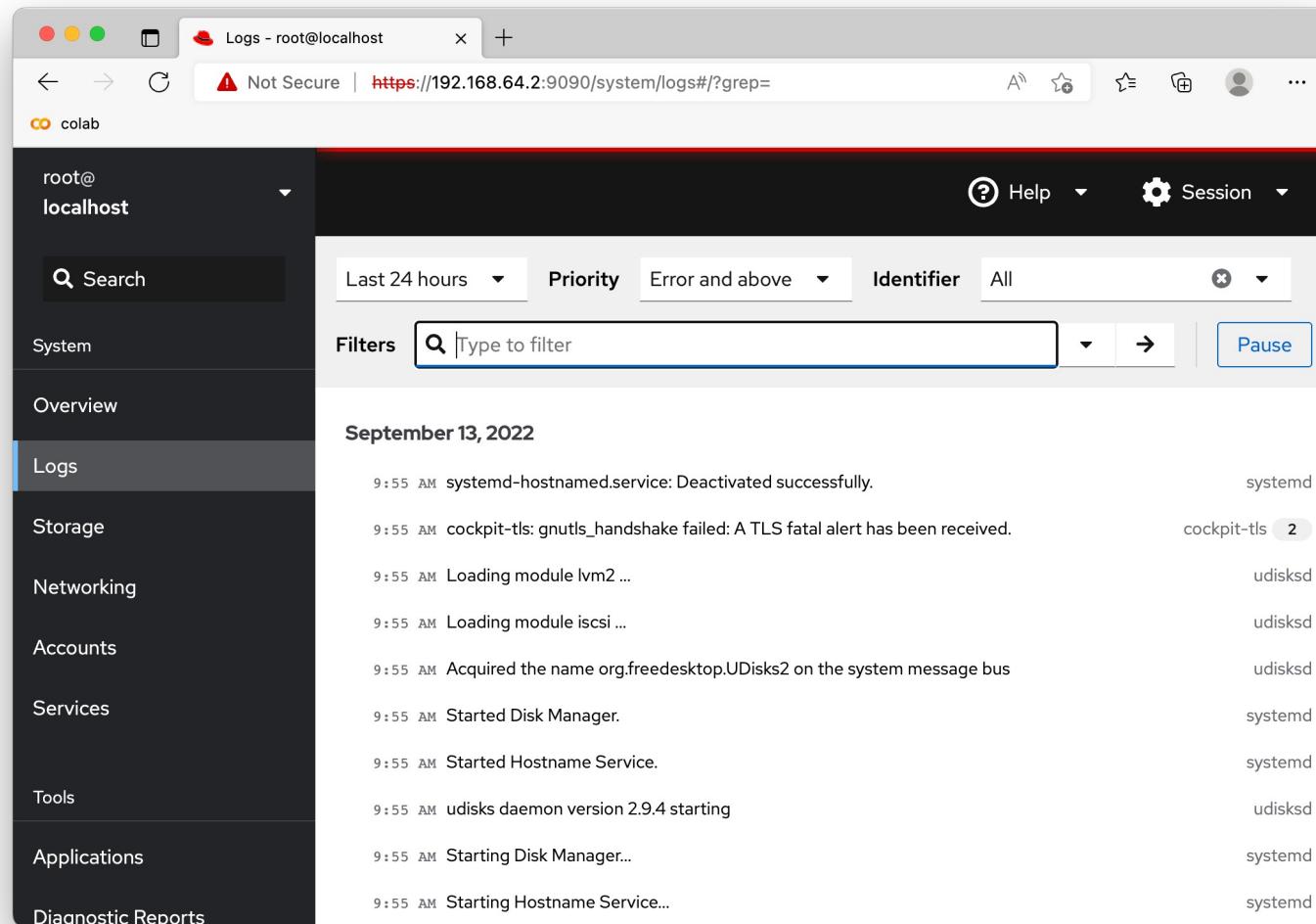
3.L Ver/cambiar el nombre del equipo

- Desde la consola web: en el menú System/Host Name
- Desde el terminal: **hostnamectl status** y **hostnamectl set-hostname nuevonombre**



3.L Mostrar los logs del sistema

- Desde la consola web: en el menú Logs
- Desde el terminal: **journalctl** (o ver el contenido de **/var/log/messages**)



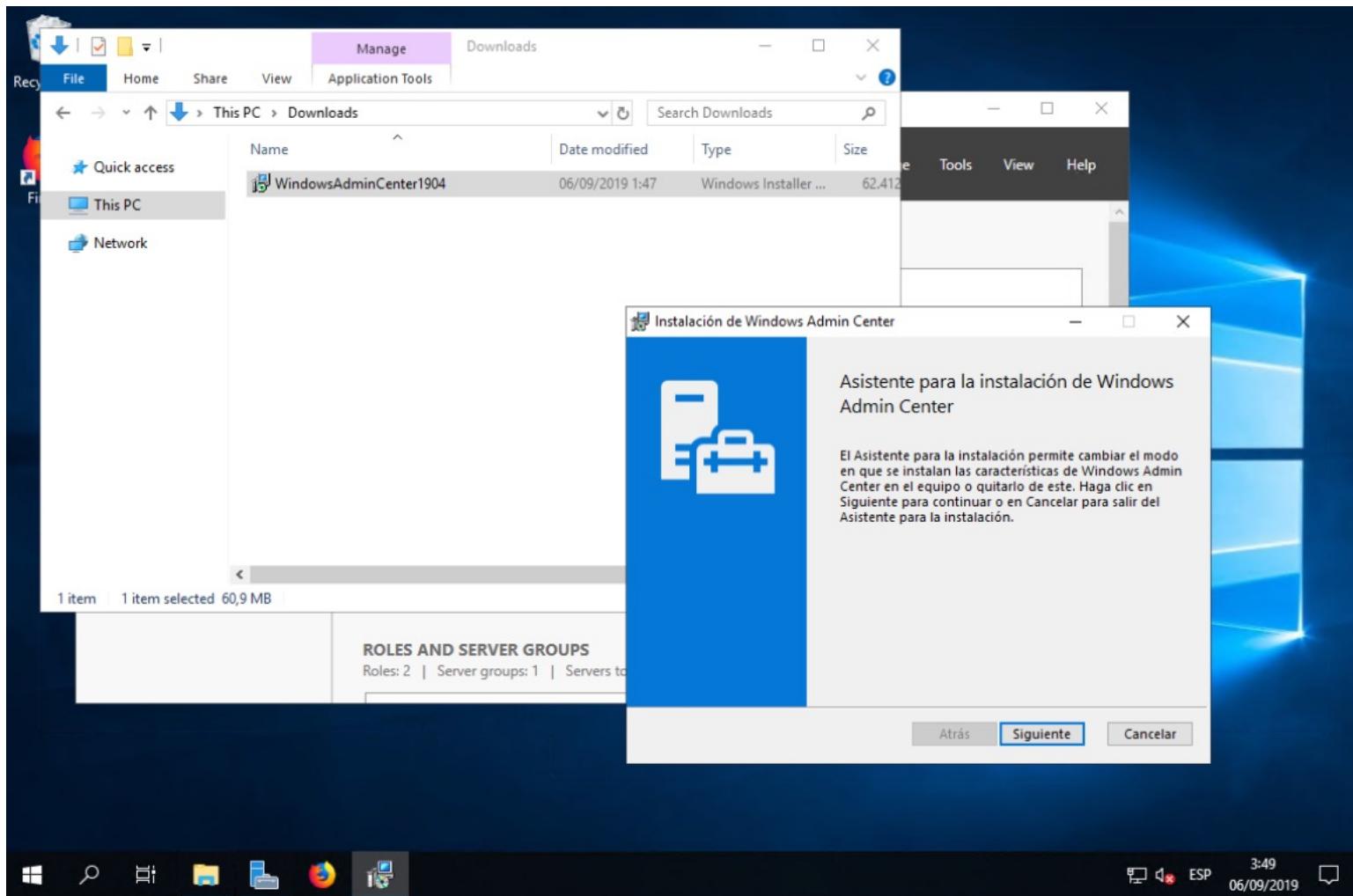
3.L Instalación de software

- Las órdenes para instalar software son más cómodas desde la terminal
- Instalación de un paquete: **yum install** paquete
- Listado de repositorios software: **yum repolist**
- Listado de paquetes que contengan una palabra: **yum search** palabra
- Actualizar un paquete: **yum update** paquete
- Actualizar todos los paquetes: **yum update**
- Eliminar un paquete y todas sus dependencias: **yum remove** paquete
- Listar todos los paquetes existentes en los repositorios: **yum list all**
- Listar todos los paquetes instalados: **yum list installed**

3.W Tareas de administración básica Windows Server

- WS puede administrarse desde una consola de **Powershell**, desde un escritorio gráfico con la herramienta **Server Manager** o desde una consola web con la herramienta **Windows Admin Server**.
- La mayoría de las tareas pueden hacerse desde el gestor web, pero este incluye además una conexión de **Remote Desktop** y una consola de Powershell, con lo que no es necesario iniciar sesión local en el equipo para administrarlo.
- Windows Admin Server debe descargarse e instalarse desde un archivo .msi (ver siguiente transparencia). El instalador configura el firewall e instala un certificado válido por 60 días.
- El servidor escucha en un puerto configurable desde el instalador. Si se redirigen las conexiones no encriptadas que se realicen al puerto 80 a conexiones https, puede accederse a Windows Admin Server con una conexión a <http://localhost> desde el servidor. Al igual que en Linux, si el servidor está virtualizado, para poder acceder desde el exterior al interfaz de red debe estar en modo **bridged** (o **host-only** si se configura adecuadamente la red en la máquina virtual).

3.W Instalación de Windows Admin Center



3.W Windows Admin Center desde Chrome

The screenshot shows the Windows Admin Center interface within a web browser window. The title bar includes "Windows Admin Center" and "Server Manager". The Microsoft logo is at the top right, along with navigation icons for back, forward, refresh, and help.

The main content area displays the "win-gm41ucljp0k" server details under the "Overview" tab. The left sidebar lists various management tools:

- Search Tools
- Overview (selected)
- Azure hybrid services
- Azure File Sync
- Backup
- Certificates
- Devices
- Events
- Files
- Firewall
- Installed Apps
- Local Users & Groups
- Network
- PowerShell
- Processes
- Registry
- Remote Desktop
- Roles & Features
- Settings

The "Overview" section contains the following information:

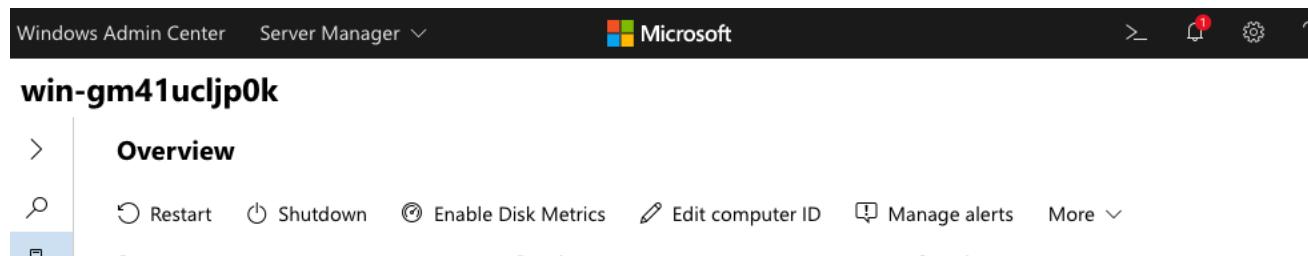
Computer name	Domain	Operating system
win-gm41ucljp0k	-	Microsoft Windows Server 2019 Standard
Version	Installed memory (RAM)	Disk space (Free / Total)
10.0.17763	6 GB	1.96 TB / 2.27 TB
Processors	Manufacturer	Model
Intel(R) Xeon(R) CPU E5530 @ 2.40GHz Intel(R) Xeon(R) CPU E5530 @ 2.40GHz	Hewlett-Packard	HP Z600 Workstation
Logical processors	Windows Defender	NIC(s)
16	Real-time protection: On	2
Azure Backup status	Up time	Logged in users
Not protected	77:24:53	1

The "CPU" section features a performance chart showing utilization over time. The chart has a Y-axis from 0 to 100 and an X-axis from "60 seconds ago" to "Now". Key metrics displayed are:

- Utilization: 0.37%
- Handles: 50660
- Speed: 1.8GHz
- Processes: 119
- Threads: 1403

3.W Apagar y reiniciar el sistema

- Desde la consola web: Indicar Restart o Shutdown
- Desde la consola powershell: **Stop-Computer** y **Restart-Computer**



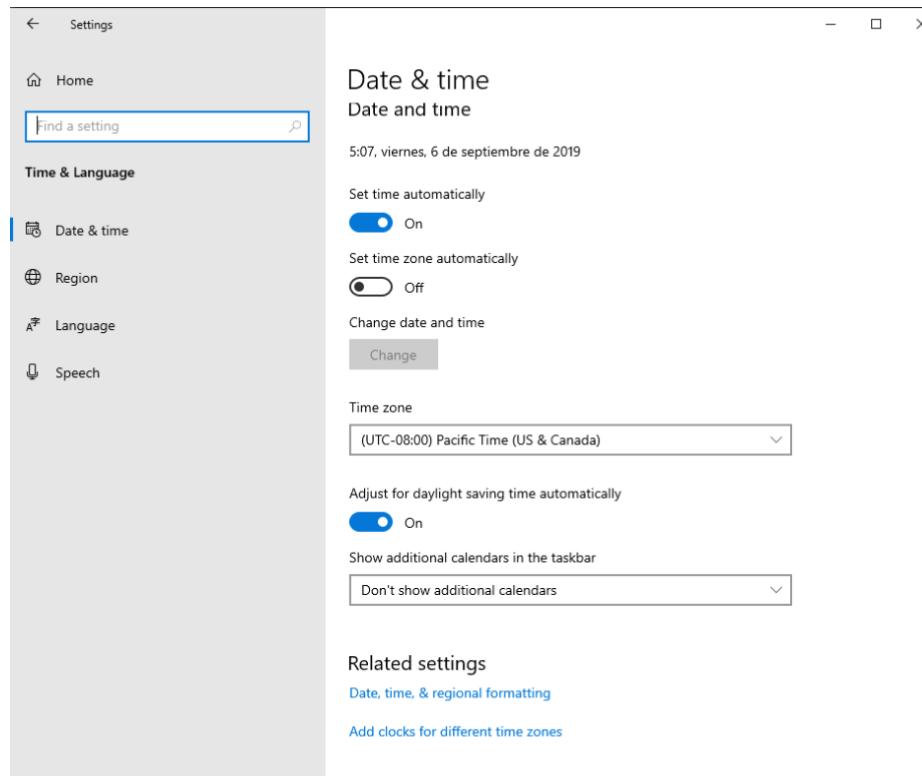
3.W Ver/cambiar el nombre del equipo

- Desde la consola web: Edit computer ID
- Desde la consola powershell: **Rename-computer**
- Si se cambia el nombre del equipo puede ser necesario volver a configurar los certificados de Windows Admin Center (ver <http://coryretherford.com/Lists/Posts/Post.aspx?ID=418>)

The screenshot shows the Windows Admin Center interface. On the left, there's a sidebar with various icons. The main area has a title bar "win-gm41ucljp0k". Below it, the "Overview" section displays system details like Computer name (win-gm41ucljp0k), Version (10.0.17763), Processors (Intel(R) Xeon(R) CPU E5530 @ 2.40GHz), Logical processors (16), and Azure Backup status (Not protected). To the right, the "Edit computer ID" page is open. It has fields for "Computer name" (WIN-GM41UCLJP0K) and "Name" (WORKGROUP). Below these, the "Membership" section shows options for "Domain" (selected) and "Workgroup" (selected), each with a "Name" field.

3.W Configuración de fecha, hora y teclado

- Desde la consola web: Settings (puede no estar accesible, dependiendo de las directivas que imponga la organización)
- Desde el escritorio: Settings
- Desde el terminal: **Get-Date**, **Set-Date**, **Get-TimeZone**, **Set-TimeZone**



3.W Mostrar los logs del sistema

- Desde la consola web: Events
- Desde el terminal: **Get-EventLog nombre** (p.e. **Get-EventLog Application** – usar **Get-EventLog *** para obtener la lista)

The screenshot shows the Windows Admin Center interface for a server named 'w2019as'. The left sidebar has a 'Tools' section with various icons and links, and the 'Events' link is currently selected, highlighted with a blue background.

The main content area is titled 'Events' and displays the 'Administrative Logs' section. It includes a tree view of log categories:

- Windows Logs
 - Application
 - Security
 - System
 - ForwardedEvents
 - Setup
- Applications and Services Logs
 - Hardware Events
 - Intel
 - Internet Explorer
 - Key Management Service
 - Microsoft
 - Microsoft-ServerManagementExperience
 - NIS
 - OpenSSH

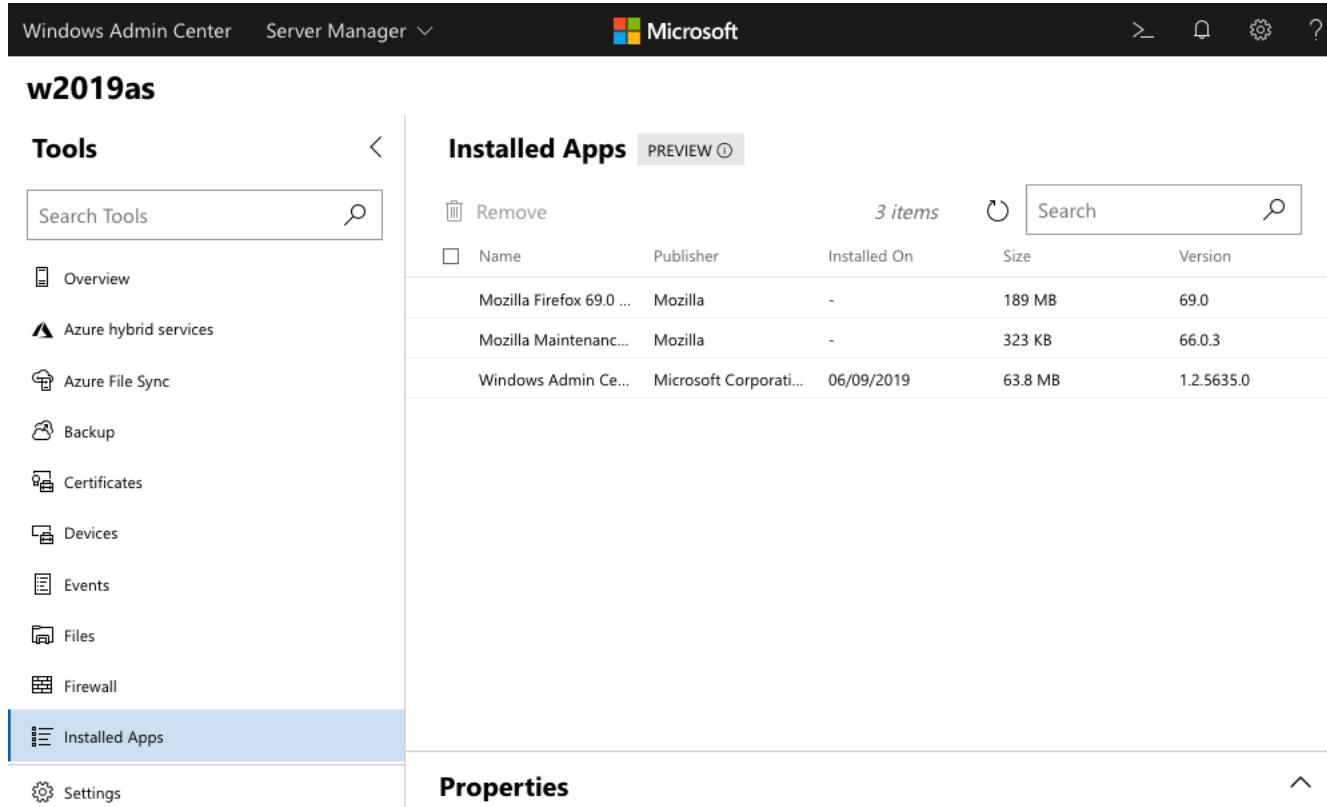
Below the tree view is a table titled 'Export' showing 934 items. The columns are Level, Date and T..., Source, and Event ID. The table lists several log entries, such as Information and Error events from Microsoft components like Microsoft-ServerManagementExperience and Internet Explorer.

Level	Date and T...	Source	Event ID
Information	06/09/2019,...	Microsoft-...	102
Information	06/09/2019,...	Microsoft-...	101
Information	06/09/2019,...	Microsoft-...	106
Information	06/09/2019,...	Microsoft-...	112
Error	06/09/2019,...	Microsoft-...	131
Information	06/09/2019,...	Microsoft-...	112
Error	06/09/2019,...	Microsoft-...	131
Information	06/09/2019,...	Microsoft-...	112
Error	06/09/2019,...	Microsoft-...	131
Information	06/09/2019,...	Microsoft-...	112

A 'Details' button is located at the bottom right of the table.

3.W Instalación de software

- Desde la consola web: Installed Apps
- Desde powershell: **Install-Package -Name ".\paquete.msi"**



The screenshot shows the Windows Admin Center interface with the title "w2019as". On the left, there's a sidebar titled "Tools" with a "Search Tools" input field. Below it are links for "Overview", "Azure hybrid services", "Azure File Sync", "Backup", "Certificates", "Devices", "Events", "Files", "Firewall", and "Installed Apps", which is currently selected and highlighted in blue. The main content area is titled "Installed Apps" and shows a table with three items:

Name	Publisher	Installed On	Size	Version
Mozilla Firefox 69.0 ...	Mozilla	-	189 MB	69.0
Mozilla Maintenanc...	Mozilla	-	323 KB	66.0.3
Windows Admin Ce...	Microsoft Corporati...	06/09/2019	63.8 MB	1.2.5635.0

3.W Instalación de software

- Para instalar roles y características

The screenshot shows the Windows Admin Center interface for managing server roles and features. The left sidebar has a 'Tools' section with various icons for PowerShell, Processes, Registry, Remote Desktop, Roles & Features (which is selected and highlighted in blue), Scheduled Tasks, Services, Storage, Storage Migration Service, Storage Replica, and Settings. The main content area is titled 'Roles and Features' and displays a table of installed and available roles. The table has columns for Name, State, and Type. Under the 'Roles' heading, there are 10 of 91 installed roles, including Active Directory Certificate, Active Directory Domain Ser..., Active Directory Federation, Active Directory Lightweight, Active Directory Rights Man..., Device Health Attestation, DHCP Server, DNS Server, Fax Server, File and Storage Services, and Host Guardian Service. Most of these roles are listed as Available. The total count shown is 267 items.

Name	State	Type
Roles	10 of 91 Installed	
Active Directory Certificate ...	0 of 6 Installed	Role
Active Directory Domain Ser...	Available	Role
Active Directory Federation ...	Available	Role
Active Directory Lightweight...	Available	Role
Active Directory Rights Man...	0 of 2 Installed	Role
Device Health Attestation	Available	Role
DHCP Server	Available	Role
DNS Server	Available	Role
Fax Server	Available	Role
File and Storage Services	1 of 12 Installed	Role
Host Guardian Service	Available	Role

4. Configuración del arranque de un servidor

- El proceso de arranque tiene diferentes etapas: inicializar el hardware, ejecutar un **bootloader**, cargar e iniciar una imagen del **kernel** y ejecutar diferentes **scripts de inicio y daemons**.
- Para cada una de esas etapas hay diferentes herramientas: **BIOS** o **UEFI** para inicializar el hardware, **GRUB**, **LILO**, **SYSLINUX** o **Loadlin** son bootloaders de Linux, **BOOTMGR** es el bootloader de Windows.

4. Firmware de un servidor: BIOS y UEFI

- Emplearemos servidores con arquitectura Intel. El firmware de los servidores más modernos está basado en **UEFI**, pero todavía (2021) hay servidores basados en **BIOS**.
- BIOS significa “**Basic Input Output System**”. Fue creado en 1975. Su función es iniciar los componentes de hardware y lanzar el sistema operativo del servidor. También gestiona la energía y los sensores de temperatura del ordenador. Está escrita en ensamblador y es código de 16 bits. Suele combinarse con discos duros particionados con **Master Boot Record (MBR)**.
- UEFI significa “**Unified Extensible Firmware Interface**”. Está escrito en C y es el sucesor de BIOS. Realiza todas las funciones de la BIOS y añade una interfaz gráfica, un sistema de inicio seguro, soporte para discos de más de 2Tb y mejora la velocidad de arranque. La UEFI puede conectarse a Internet para actualizarse. El código de la UEFI se ejecuta en 32 o 64 bits. Suele combinarse con discos duros particionados con tabla de particiones **GUID (GPT)**.
- Normalmente se emplea la palabra “BIOS” para referirse al firmware del servidor, sea BIOS o UEFI, y la mayoría de los servidores actuales tiene los dos modos: BIOS normal (legacy) o UEFI.

4. Arranque de un servidor con BIOS

- Arranque de un servidor Intel/AMD con BIOS
 - La CPU recibe un reset
 - Se inicia la ejecución en 0xffffffff0.
 - Se salta al “reset vector”, código en memoria no volátil (BIOS) cerca del límite del mapa de memoria direccionable en 32 bits
 - El procesador comienza en modo real:
 - Espacio de direcciones de 20 bits (las 12 líneas superiores del bus de direcciones se mantienen subidas)
 - Acceso directo a E/S, interrupciones y memoria

4. Arranque de servidor con BIOS

- BIOS ejecuta:
 - Power-on self-test (**POST**)
 - Detecta la BIOS de la tarjeta de video y ejecuta la inicialización del video
 - Detecta las BIOS de otros dispositivos y las inicializa
 - Muestra la pantalla de inicio
 - Breve test de memoria
 - Configuración de memoria y parámetros del disco
 - Configuración de dispositivos Plug & Play: PCIe, USB, SATA, SPI
 - Se asignan recursos (Canales DMA, IRQs)
 - Se identifica el dispositivo de arranque
 - Se carga el bloque 0 (**Master Boot Record**) en la dirección 0x7c00 y se salta a esa dirección

4. Arranque de un servidor con UEFI

- A partir de 2005 se introduce el Unified Extensible Firmware Interface (UEFI)
- Soporta arquitecturas de 32 y de 64 bits
- No necesita poner al procesador en modo real (modo 8086 de 16 bits, con direccionamiento de 20 bits)
- El manejo de energía (**Advanced Configuration and Power Interface, ACPI**) de la BIOS se mantiene, además de otros componentes de administración del sistema presentes en la BIOS
- Se pueden utilizar drivers de periféricos en EFI Byte Code (independientes del procesador)
- No se necesita que haya un **bootloader** dedicado (aunque normalmente lo hay): los ficheros de arranque se almacenan en una partición dedicada (**EFI boot partition**)
- Se pueden cargar extensiones en memoria no volátil

4. Arranque de servidor con UEFI

- No se necesita código en el **MBR** (se ignora el bloque 0)
- Se lee la tabla de particiones **GUID (GPT)** de los bloques 1 al 33
- La EFI entiende los sistemas de archivos **FAT** de Microsoft, por lo que el kernel puede estar almacenado como un fichero (no como bloques correlativos) en la partición de sistema EFI:
 - Windows 64 bit: **BOOTMGR (Windows Boot Manager)** está en la partición EFI
 - NT (IA-64): IA64ldr
 - Linux: elilo.efi (**ELILO** es **EFI Linux Boot Loader**) o **grub (Gnu GRand Unified Bootlader)**
 - OSX: boot.efi
 - etc.

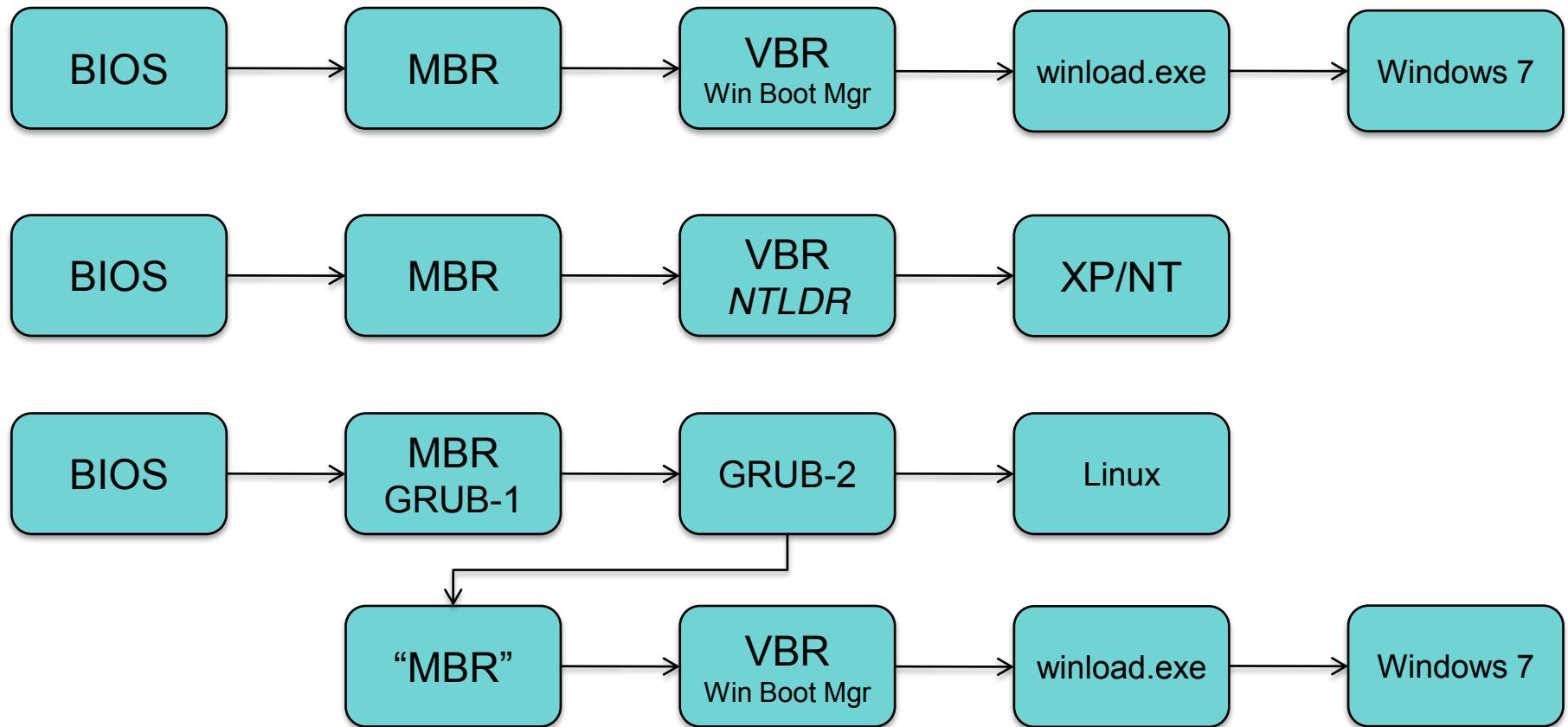
4.L Inicio de Linux en equipos con BIOS

- La BIOS carga el MBR y ejecuta el código de arranque desde el dispositivo seleccionado, que contiene la fase 1 del bootloader. La fase 1 carga la fase 2, con la mayoría del código. Algunos cargadores emplean una fase intermedia (fase 1.5) con código adicional para leer discos no soportados por la BIOS.
- El bootloader presenta un menú con diferentes opciones. Lee, de varios archivos en el disco, la **imagen del kernel** y de **dracut**, los descomprime en memoria, configura las funciones de manejo de hardware y memoria y cede el control al kernel.
- El código del kernel configura las interrupciones, manejador de memoria, periféricos, etc. Se lanzan el proceso **idle**, el **scheduler** y el proceso **systemd**.
- El proceso **systemd** configura los servicios que permiten crear un entorno de usuario y presenta la pantalla de **login**.

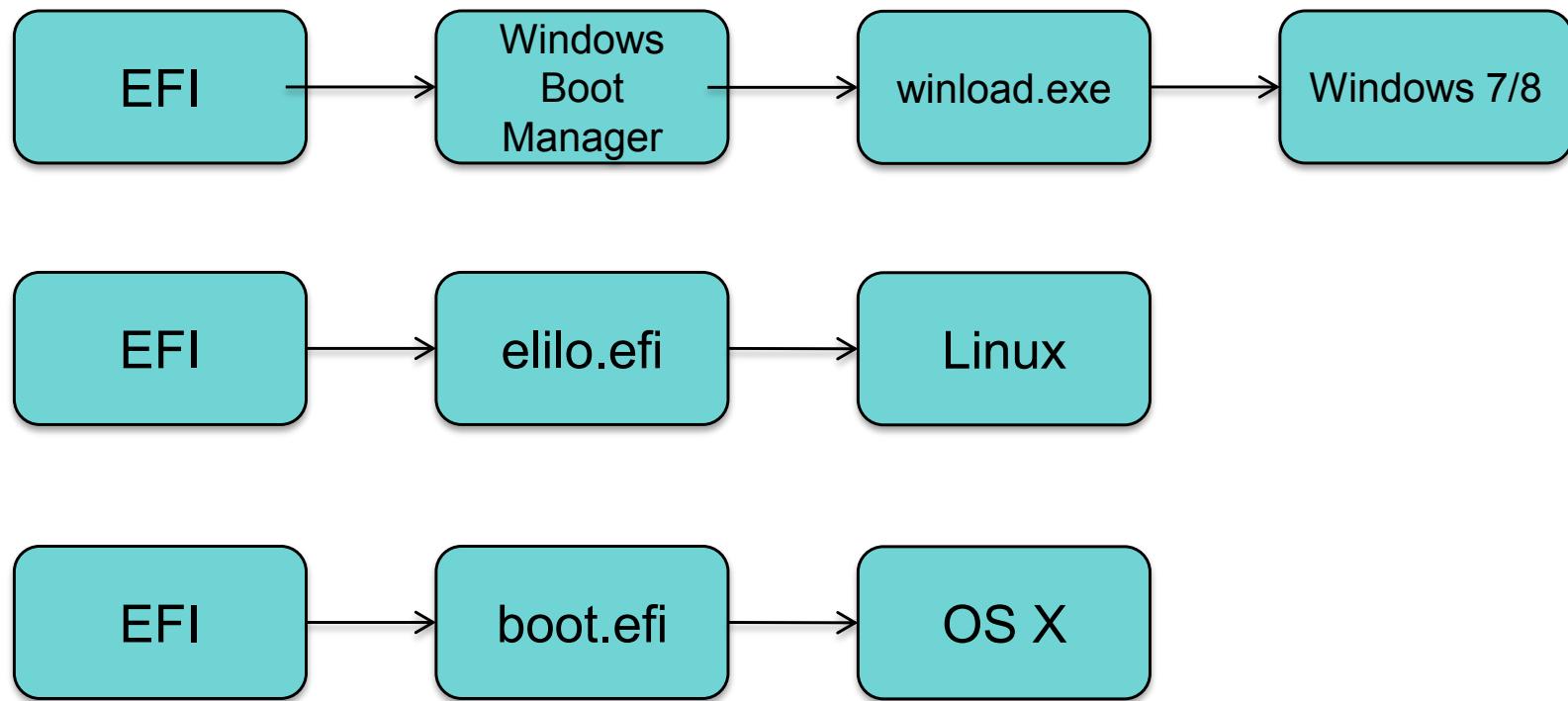
4.W Inicio de Windows en equipos con BIOS

- La BIOS no entiende los sistemas de archivos pero puede leer bloques del disco
- El MBR (Master Boot Record) es el bloque 0 del disco (512 bytes)
 - Pequeño **chainloader** (sobre 440 bytes)
 - Firma del disco (4 bytes)
 - Tabla de particiones (16 bytes por partición * 4 particiones)
- El firmware del BIOS carga y ejecuta el contenido del MBR
- El código del MBR lee la tabla de particiones y carga el **Volume Boot Record (VBR)** de la partición de inicio
 - Se identifica el tipo y tamaño de la partición
 - El VBR contiene el **Instruction Program Loader (IPL)**, que ejecuta el código de inicio
 - El IPL lee sectores adicionales para cargar **BOOTMGR** (desde Windows 7) o **NTLDR** (se usaba en Windows NT, XP, 2003)

4 Arranque de un PC con BIOS: resumen



4 Arranque de un PC con UEFI: resumen



5. Arranque de un servidor desde un DVD o un lápiz USB

- Los servidores pueden venir sin un sistema operativo preinstalado en el disco duro, o el disco puede estar corrompido y no ser posible el arranque.
- En BIOS/UEFI hay un menú “Boot Order” que permite elegir el dispositivo desde el que se arranca. La forma más habitual de iniciar un servidor sin sistema operativo instalado es un lápiz USB. Algunos servidores aún emplean DVDs. También es posible botar en red con **TFTP** de un servidor **PXE** (Preboot Execution Environment).
- En algunos casos hay que deshabilitar el **Secure Boot** de la UEFI y el **Fast Startup** (el Fast Startup es un modo híbrido de hibernación, que puede bloquear algunas funciones de la BIOS)
- Frecuentemente se descarga la imagen del operativo de internet, y se crea el medio de instalación con un programa que depende del ordenador en que se haya bajado la imagen. Si se descarga una imagen de Windows desde un equipo con ese mismo operativo, puede emplearse el **Microsoft Media Creation Tool**. Si se descarga una imagen iso desde una máquina Linux puede emplearse la orden **dd**. Para crear un USB Linux botable desde una máquina Windows hay herramientas gratuitas, como **Fedora Media Writer**. Consultar <https://wiki.centos.org/HowTos/InstallFromUSBkey>

6.L Bootloaders

- Los kernel tienen que soportar numerosos periféricos y si este código estuviese compilado de forma estática en el kernel su tamaño sería inmanejable. Los kernel Linux cargan ciertos **módulos** del disco durante el arranque, en función del hardware presente en el servidor (el mismo concepto que los drivers de Windows).
- Para que el kernel pueda cargar módulos de disco es necesario que el driver de ese disco esté compilado de forma estática o bien que se proporcione al kernel la imagen de un **disco RAM** con los binarios de los módulos y que en el kernel esté compilado el soporte para ese disco RAM.
- El bootloader se lee del disco mediante la BIOS/UEFI y permite elegir la imagen de kernel que se desea botar, la imagen del disco RAM con los drivers y las diferentes opciones de arranque que se le pasan al kernel como argumentos (arranque en modo seguro, en modo single-user, etc.)

6.L Parámetros del kernel y mensajes al botar

- Algunos de los parámetros que se le pueden pasar al kernel son:
 - **root=**: monta la partición indicada como sistema de archivos raíz
 - **ro**: monta el dispositivo raíz como sólo lectura
 - **initrd=**: localización del disco RAM de inicio
 - **systemd.unit=**: bota a ese target de systemd
- Hay una lista completa en http://files.kroah.com/lkn/lkn_pdf/ch09.pdf
- La orden **journalctl --dmesg** lista los mensajes producidos en el último arranque

6.L INItial Ram Disk: initrd

- initrd es una de las tecnologías empleadas para construir el disco RAM que se le proporciona al kernel antes de que se monte el disco raíz,
- Necesidad:
 - Drivers para **RAID, LVM, NFS**, etc.
 - Soporte para **hibernación** (acceso al archivo con el contenido de la RAM)
- Implementación
 - Pequeño disco RAM en **/dev/ram** montado como filesystem al botar
 - Driver compilado de forma estática en el kernel
 - **Ext2 o cramfs** (compressed ROM file system, puede usarse sin descomprimir)
- Configuración
 - **mkinitrd**

6.L Contenido de una imagen initrd

- ```
gunzip -c /boot/initrd-2.6.31.img > /tmp/my_initrd
$ cpio -it < /tmp/my_initrd [examine contents]
lib
lib/udev
lib/udev/console_init
lib/firmware
lib/firmware/radeon
lib/firmware/radeon/RV730_me.bin
... snip ...
lib/modules
lib/modules/2.6.31
lib/modules/2.6.31/radeon.ko
lib/modules/2.6.31/modules.isapnpmap
... and so on and so on ...
$ cd [somewhere] [in preparation for unloading]
$ cpio -i < /tmp/my_initrd [unload]
```

## 6.L INITial RAM File System: initramfs

---

- **initramfs** es un sucesor de initrd
- Archivo **cpio** que se copia directamente en memoria
- Basado en **ramfs** o **tmpfs**
- Es más eficiente que initrd
- No requiere drivers de sistemas de archivos compilados en el kernel
- No requiere /dev/ram
- Se crea con **dracut**. Se genera automáticamente cada vez que se actualiza el kernel. Normalmente se guardan las dos últimas versiones de kernel e initramfs
- Su contenido puede verse con **lsinitrd**

## 6.L dracut

---

- El comando dracut se usa para modificar el contenido del initramfs. Se basa en eventos generados por **udev**.
- Para regenerar un initramfs con sólo los módulos necesarios en el sistema actual, se emplea la orden **dracut --force**.
- Si se desea crear un disco RAM con todos los drivers (por ejemplo, si se va a mover el disco de inicio a otro servidor con diferente hardware) se ejecuta el comando **dracut --force --no-hostonly**

## 6.L udev

---

- Udev es un **daemon** que crea, nombra y destruye nodos de dispositivo (las entradas en el directorio /dev). Se usa principalmente para dar persistencia a los nombres de los dispositivos.
- También maneja **eventos** en el espacio de usuario que se lanzan cuando se añade o se retira hardware del equipo (p.e. pinchar un lápiz USB, cerrar la tapa de un portátil, etc.)
- **udevd** maneja:
  - Dispositivos USB -> udisksd-daemon
  - Conexión/desconexión del cable Ethernet -> NetworkManager-daemon -> dhclient
- Reemplaza a HAL (Hardware Abstraction Layer)
- Configuración en **/etc/udev**

# 6.L Gestor de arranque Grub 1 (legacy)

---

- RedHat Linux (hasta la versión 6) utilizaba el boot loader GRUB. La configuración de GRUB estaba en /boot/grub/grub.conf, archivo donde se listaban todos los sistemas operativos disponibles y sus parámetros de arranque.

```
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Fedora Core (2.6.8-1.521)
 root (hd0,0)
 kernel /vmlinuz-2.6.8-1.521 ro root=LABEL=/
 initrd /initrd-2.6.8-1.521.img
title Windows 2000
 rootnoverify (hd0,1)
 chainloader +1
```

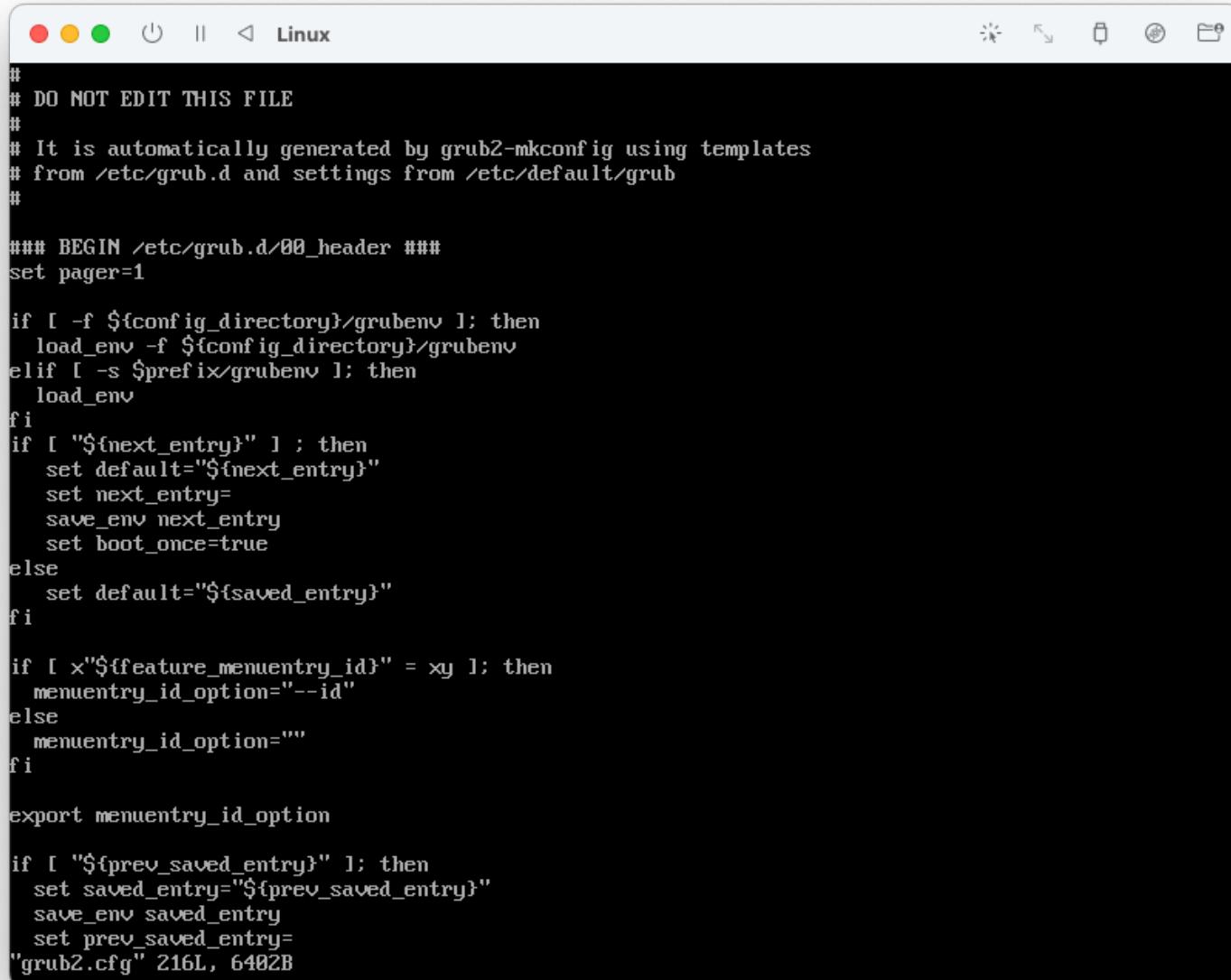
## 6.L Gestor de arranque GRUB 2

---

- La versión 2 de grub es la opción por defecto a partir de la versión 7
- Los ficheros de configuración están en **/etc/grub2.cfg**, **/etc/grub2-efi.cfg**, **/etc/default/grub** y el directorio **/etc/grub.d/**
  - grub2.cfg o grub2-efi.cfg se generan automáticamente y no se editna a mano
  - el directorio grub.d contiene los scripts GRUB que se usan para construir grub2.cfg (grub2.cfg es una concatenación de estos archivos)
  - /etc/default/grub contiene opciones que definen cómo se procesan los archivos de /etc/grub.d/ para generar grub2.cfg

# 6.L /etc/grub2.cfg

---



```

DO NOT EDIT THIS FILE

It is automatically generated by grub2-mkconfig using templates
from /etc/grub.d and settings from /etc/default/grub

BEGIN /etc/grub.d/00_header ###
set pager=1

if [-f ${config_directory}/grubenv]; then
 load_env -f ${config_directory}/grubenv
elif [-s $prefix/grubenv]; then
 load_env
fi
if ["${next_entry}"] ; then
 set default="${next_entry}"
 set next_entry=
 save_env next_entry
 set boot_once=true
else
 set default="${saved_entry}"
fi

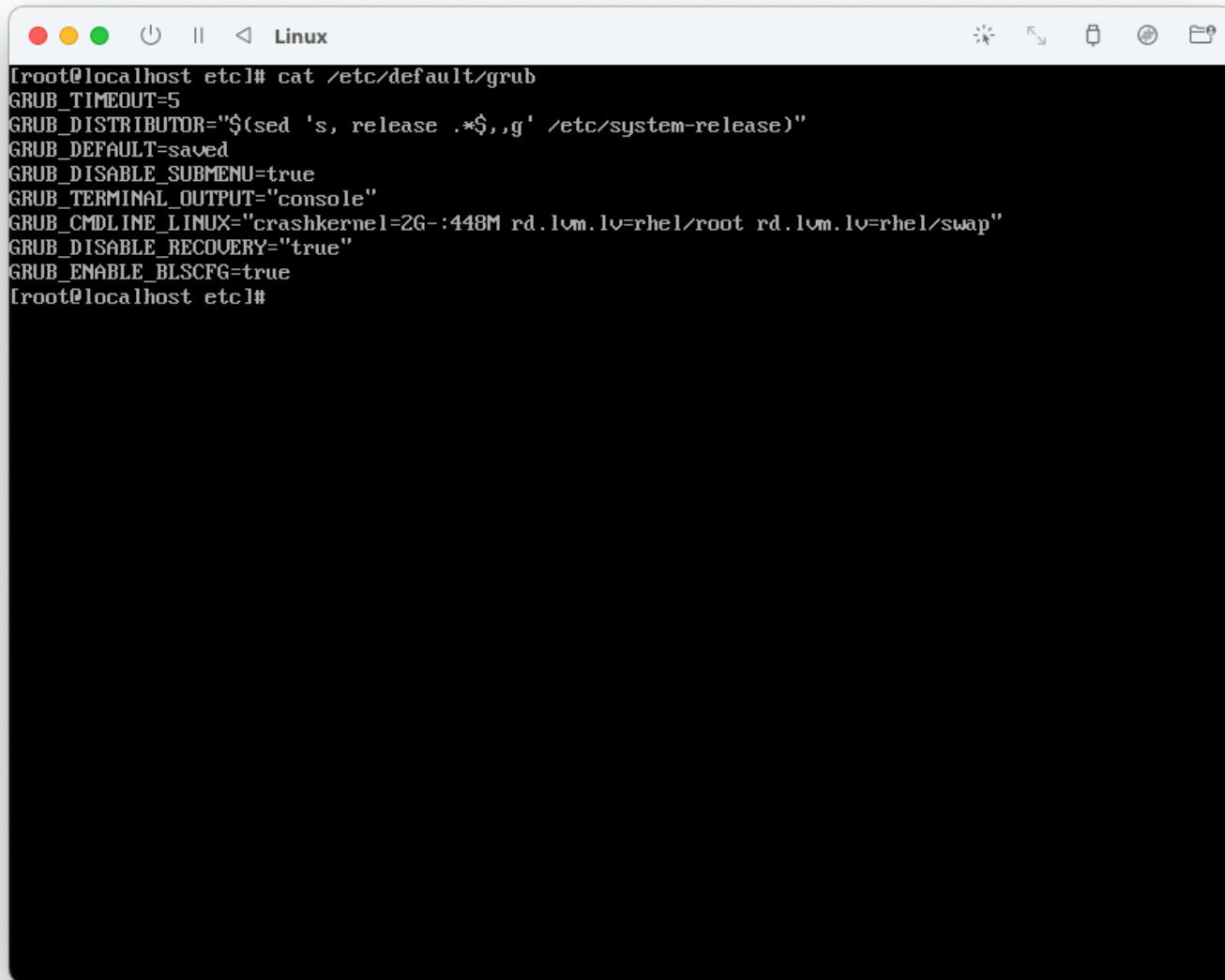
if ["x${feature_menuentry_id}" = xy]; then
 menuentry_id_option="--id"
else
 menuentry_id_option=""
fi

export menuentry_id_option

if ["${prev_saved_entry}"]; then
 set saved_entry="${prev_saved_entry}"
 save_env saved_entry
 set prev_saved_entry=
"grub2.cfg" 216L, 6402B
```

# 6.L /etc/default/grub

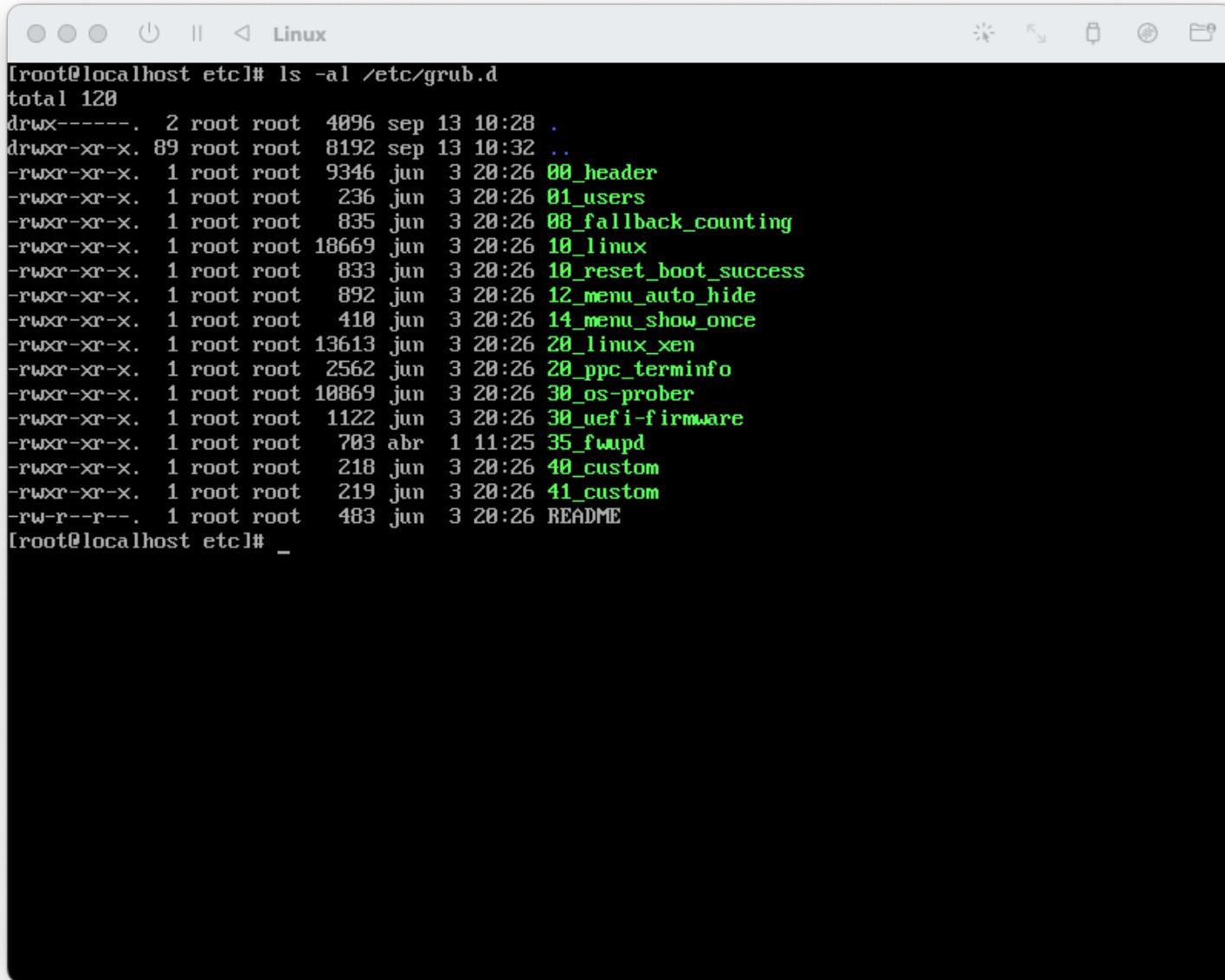
---



A screenshot of a Linux terminal window titled "Linux". The window has a dark background and light-colored text. It displays the contents of the file "/etc/default/grub" as follows:

```
[root@localhost etc]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*\$, ,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=2G-:448M rd.lom.lv=rhel/root rd.lom.lv=rhel/swap"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
[root@localhost etc]#
```

# 6.L Gestor de arranque Grub 2



```
[root@localhost etc]# ls -al /etc/grub.d
total 120
drwx----- 2 root root 4096 sep 13 10:28 .
drwxr-xr-x 89 root root 8192 sep 13 10:32 ..
-rwxr-xr-x 1 root root 9346 jun 3 20:26 00_header
-rwxr-xr-x 1 root root 236 jun 3 20:26 01_users
-rwxr-xr-x 1 root root 835 jun 3 20:26 08_fallback_counting
-rwxr-xr-x 1 root root 18669 jun 3 20:26 10_linux
-rwxr-xr-x 1 root root 833 jun 3 20:26 10_reset_boot_success
-rwxr-xr-x 1 root root 892 jun 3 20:26 12_menu_auto_hide
-rwxr-xr-x 1 root root 410 jun 3 20:26 14_menu_show_once
-rwxr-xr-x 1 root root 13613 jun 3 20:26 20_linux_xen
-rwxr-xr-x 1 root root 2562 jun 3 20:26 20_ppc_terminfo
-rwxr-xr-x 1 root root 10869 jun 3 20:26 30_os-prober
-rwxr-xr-x 1 root root 1122 jun 3 20:26 30_uefi-firmware
-rwxr-xr-x 1 root root 703 abr 1 11:25 35_fwupd
-rwxr-xr-x 1 root root 218 jun 3 20:26 40_custom
-rwxr-xr-x 1 root root 219 jun 3 20:26 41_custom
-rw-r--r-- 1 root root 483 jun 3 20:26 README
[root@localhost etc]# -
```

## 6.L Gestor de arranque Grub 2

---

- **00\_header** carga la configuración desde /etc/default/grub: timeout, orden de las entradas, etc.
- **01\_users** sirve para establecer una contraseña en el bootloader (el comando grub2-setpassword genera un archivo /boot/grub2/user.cfg con el símbolo GRUB2\_PASSWORD definiendo la password encriptada)
- **08\_fallback\_counting** sirve para que el sistema bote de un nuevo kernel si la opción por defecto falla
- **10\_linux** busca kernels instalados y los añade al menú de boot
- **30\_os-prober/30\_uefi-firmware** se usan para configurar grub en nuevas instalaciones
- **40\_custom** es un patrón que se usa para añadir entradas adicionales al menú de boot

# 6.L Gestor de arranque Grub 2

---

- **/etc/default/grub** contiene la configuración adaptada. **/etc/grub.d/** contiene archivos con la información del menú y los scripts de arranque.
- Cuando se ejecuta el comando **grub2-mkconfig -o /etc/grub2.cfg** se lee el contenido de estos archivos y se genera **/etc/grub2.cfg** (o **/etc/grub2-efi.cfg**)
- El script **/etc/grub.d/10\_linux** busca los diferentes kernels que estén instalados en la misma partición. El script **/etc/grub.d/30\_os-prober** busca otros sistemas operativos. También se añaden automáticamente entradas al menú de grub cada vez que se actualiza el kernel.
- El script **40\_custom** es un patrón para añadir entradas hechas a la medida:

```
#!/bin/sh
exec tail -n +3 $0
This file provides an easy way to add custom menu entries. Simply type the
menu entries you want to add after this comment. Be careful not to change
the 'exec tail' line above.
~
```

# 6.L Gestor de arranque Grub 2

- Una entrada de menú en el fichero **40\_custom** tiene la forma mostrada en la figura inferior. **set root** apunta a la partición **/boot**. El parámetro **root** que se le pasa a al kernel apunta a la partición **/**

# 6.L Gestor de arranque Grub 2

---

- En RHEL9 no hay un menuentry por cada kernel instalado; se usa el framework BLS y se generan automáticamente estas entradas desde **/boot/loader/entries**
- En el archivo **/etc/default/grub**, el símbolo **GRUB\_DEFAULT** contiene por defecto la palabra **saved**. Esto le indica a GRUB 2 que cargue el kernel especificado por la directiva **saved\_entry** en **/boot/grub2/grubenv**. **saved\_entry** apunta al último kernel instalado, y se puede cambiar con **grub2-set-default**

```
[root@localhost grub2]# cd /boot/grub2
[root@localhost grub2]# ls -al
total 16
drwx----- 3 root root 50 sep 13 10:36 .
dr-xr-xr-x 6 root root 4096 sep 13 10:31 ..
drwx----- 2 root root 25 sep 13 10:28 fonts
-rwx----- 1 root root 6402 sep 13 10:29 grub.cfg
-rw-r--r-- 1 root root 1024 sep 13 10:29 grubenv
[root@localhost grub2]# cat grubenv
GRUB Environment Block
WARNING: Do not edit this file by tools other than grub-editenv!!!
saved_entry=e55caf9a2ed4a0cbc50bd9a2c0cc3a0-5.14.0-70.22.1.e19_0.aarch64
boot_success=0
```

## 6.L Gestor de arranque Grub 2

---

- Para botar un windows se encadenan los bootloaders. Grub carga un archivo con el contenido del sector de arranque de la partición windows, y ejecuta su código, con lo que se carga el bootloader de windows con sus propias opciones

```
#!/bin/sh -e
echo "Chainload windows"
cat << EOF
menuentry "Mi windows" {
 set root=(hd0,2)
 chainloader (hd0,1)+1
}
EOF
```

## 6.L Gestor de arranque Grub 2

---

- La orden **grub2-install** reinstala grub2 en el sector de arranque si este se ha modificado (por ejemplo, si se ha cambiado el disco o se ha instado un windows después del linux)
- **grub2-install /dev/sda** instala grub en el sector de arranque del disco
- **grub2-install /dev/sda2** instala grub en el sector de arranque de la segunda partición (últil si se desea enlazar dos bootloaders)
- En equipos con UEFI no es necesario instalar grub en el sector de arranque. El directorio `/boot/efi/EFI/redhat/` contiene `grub.efs`, que es una versión de grub compilada como una aplicación EFI para la arquitectura del firmware correspondiente. El boot manager de la EFI selecciona `grub.efs` como bootloader y lo carga en memoria; sólo es necesario generar el archivo de configuración en la partición EFI. Normalmente tampoco es necesario hacer chainloading en equipos con EFI.
- **grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg**

# 6.L Secuencia de arranque (después del bootloader)

---

- Una vez que se ha cargado el kernel y la imagen del disco RAM, el kernel toma control de la máquina y ejecuta el proceso **systemd**, que realiza las siguientes tareas de inicialización:
- Se chequea el sistema de archivos raíz con **fsck**
  - Si al apagar el sistema se desmontó correctamente el sistema de ficheros, no se chequea, aunque cada cierto número de arranques o de un tiempo determinado se chequeará de todos modos.
  - Si fsck encuentra problemas que no puede solucionar sin intervención del operador, lleva al sistema a modo monousuario.
- Se monta en sistema de ficheros raíz en modo lectura-escritura
- Se chequea el resto de sistemas de ficheros con fsck

## 6.L Secuencia de arranque (continuación)

---

- Se monta el resto de sistemas de ficheros
- Se activan las particiones de **swap** con **swapon -a**
- Se activan las quotas de disco: **quotacheck -a** y **quotaon -a**
- Se lanzan los servicios:
  - **bdflush** y **update** (que guardan los bloques de disco modificados en memoria)
  - **crond, atd**
  - **cups**
  - **syslogd**

## 6.L Secuencia de arranque (continuación)

---

- Se activa la red
- Se lanzan los servicios de red (xinetd, sendmail, named, routed, portmap, nfsd, rpc.statd, ypbind, ypserv, etc.)
- Se limpian los directorios temporales /tmp, etc.
- Se autoriza el login de usuarios
  - mediante mingetty en modo texto, en la consola
  - en modo gráfico, si se ha activado

## 7.L Inicio y detención de servicios

---

- Algunos de los servicios proporcionados por el equipo se inician automáticamente cuando este se inicia. En general, hay diferentes modos de funcionamiento, desde el modelo **single user** (donde solamente el administrador está en el sistema) hasta los modos multiusuario sin y con interfaz gráfica.
- En los sistemas unix antiguos (**SysV**, hasta RHEL 6) se definen 7 **runlevels**. Cada runlevel contiene una selección de servicios de sistema. El administrador del sistema define en qué runlevel funciona la máquina.
- Desde RHEL 7 los runlevel se han reemplazado por **targets**. Cada target es un archivo con la extensión **.target**, cuyo propósito es agrupar diferentes unidades de systemd mediante una cadena de dependencias. Por ejemplo, la unidad **graphical.target** inicia servicios del sistema como **GNOME Display Manager (gdm.service)**, el gestor de cuentas de usuario (**accounts-daemon.service**) y activa la unidad **multi-user.target**, que inicia a su vez servicios esenciales con **NetworkManager (NetworkManager.service)** o **D-Bus (dbus.service)**, y activa a su vez otro target llamado **basic.target**
- Por compatibilidad, desde RHEL 7 se proporcionan targets con una funcionalidad similar a los runlevels de SysV.

# 7.L Runlevels SysV (legacy)

---

| Modo | Directorio  | Descripción            |
|------|-------------|------------------------|
| 0    | /etc/rc0.d/ | Halt                   |
| 1    | /etc/rc1.d/ | Single User            |
| 2    | /etc/rc2.d/ | No usado               |
| 3    | /etc/rc3.d/ | Multiusuario (no GUI)  |
| 4    | /etc/rc4.d/ | No usado               |
| 5    | /etc/rc5.d/ | Multiusuario (con GUI) |
| 6    | /etc/rc6.d/ | Reboot                 |

- En máquinas SysV, el kernel ejecuta en primer lugar /sbin/init, que hace algunos chequeos del sistema: verifica la integridad de los sistemas de archivos, y arranca varios procesos.
- A continuación se inspecciona el fichero /etc/inittab para determinar el runlevel en que el sistema operará.

## 7.L Runlevels SysV (legacy)

---

```
[root@bigboy tmp]# ls /etc/rc.d/rc3.d
... ... K75netfs K96pcmcia ...
... ... K86nfsllock S05kudzu ...
... ... K87portmap S09wlan ...
... ... K91isdn S10network ...
... ... K92iptables S12syslog ...
... ... K95firstboot S17keytable ...
[root@bigboy tmp]#
```

- En función del runlevel seleccionado, el proceso init ejecuta los scripts de arranque localizados en los subdirectorios /etc/rc.d/rc0.d hasta /etc/rc.d/rc6.d, respectivamente
- Cada uno de los ficheros en esos directorios comienza con una “S”, lo que significa que se ejecuta durante el arranque (startup) o una “K”, y se ejecuta durante la parada (kill).

## 7.L Runlevels SysV (legacy)

---

```
[root@bigboy tmp]# ls /etc/rc.d/rc3.d
... ... K75netfs K96pcmcia ...
... ... K86nfslock S05kudzu ...
... ... K87portmap S09wlan ...
... ... K91isdn S10network ...
... ... K92iptables S12syslog ...
... ... K95firstboot S17keytable ...
[root@bigboy tmp]#
```

- RHEL6 almacenaba todos los scripts en el directorio /etc/init.d y los directorios /etc/rc.d contienen enlaces simbólicos al fichero correspondiente.
- Si se elimina un fichero de /etc/rc.d no se elimina el fichero de /etc/init.d
- El número que sigue a K o a S determina el orden (ascendente) en que se ejecuta el script. Por ejemplo, kudzu, con un valor 05, se ejecuta antes que wlan, con un valor 09.
- Hay una utilidad chkconfig que crea los enlaces correctos cuando se instala, activa o desactiva un servicio

# 7.L Runlevels SysV (legacy)

---

```
Default runlevel. The runlevels used by RHS are:
0 - halt (Do NOT set initdefault to this)
1 - Single user mode
2 - Multiuser, without NFS (The same as 3, if you do not have networking)
3 - Full multiuser mode
4 - unused
5 - X11
6 - reboot (Do NOT set initdefault to this)

id:3:initdefault: # Console Text Mode
id:5:initdefault: # Console GUI Mode
```

- El runlevel por defecto se define en el fichero /etc/inittab, con la variable initdefault. Cuando se define a 3, el sistema bota con el interfaz de texto en la consola. Cuando se define a 5, se bota con interfaz gráfico.

# 7.L Arrancado y Parada de Servicios SySV (legacy)

---

```
root@u-bigboy:~# /etc/init.d/apache start
 * Starting apache 1.3 web server...
 ...done.

root@u-bigboy:~#
```

- Para arrancar un servicio se usa su script en el directorio /etc/rc.d, dándole como argumento la keyword “start”
- Para pararlo, se usar la keyword “stop”
- Para rearrancar un servicio (por ejemplo, tras cambiar su configuración) se usa la keyword “restart”..
- El comando “service” sirve para lo mismo: service httpd stop. Este comando también permite una keyword “status”, que produce un breve informe.

```
[root@bigboy ~]# service httpd status
httpd (pid 6135 6133 6132 6131 6130 6129 6128 6127 1561) is running...
[root@bigboy ~]#
```

# 7.L chkconfig SySV (legacy)

---

```
[root@bigboy tmp]# chkconfig --list
keytable 0:off 1:on 2:on 3:on 4:on 5:on 6:off
atd 0:off 1:off 2:off 3:on 4:on 5:on 6:off
syslog 0:off 1:off 2:on 3:on 4:on 5:on 6:off
gpm 0:off 1:off 2:on 3:on 4:on 5:on 6:off
kudzu 0:off 1:off 2:off 3:on 4:on 5:on 6:off
wlan 0:off 1:off 2:on 3:on 4:on 5:on 6:off
sendmail 0:off 1:off 2:off 3:on 4:off 5:on 6:off
netfs 0:off 1:off 2:off 3:on 4:on 5:on 6:off
network 0:off 1:off 2:on 3:on 4:on 5:on 6:off
random 0:off 1:off 2:on 3:on 4:on 5:on 6:off
...
...
```

- El comando chkconfig puede usarse para definir qué aplicaciones corren en cada runlevel. Crea los enlaces simbólicos correspondientes en /etc/rc.d. Por ejemplo, para que sendmail no arranque en los runlevels 3 y 5 hacemos chkconfig --level 35 sendmail off

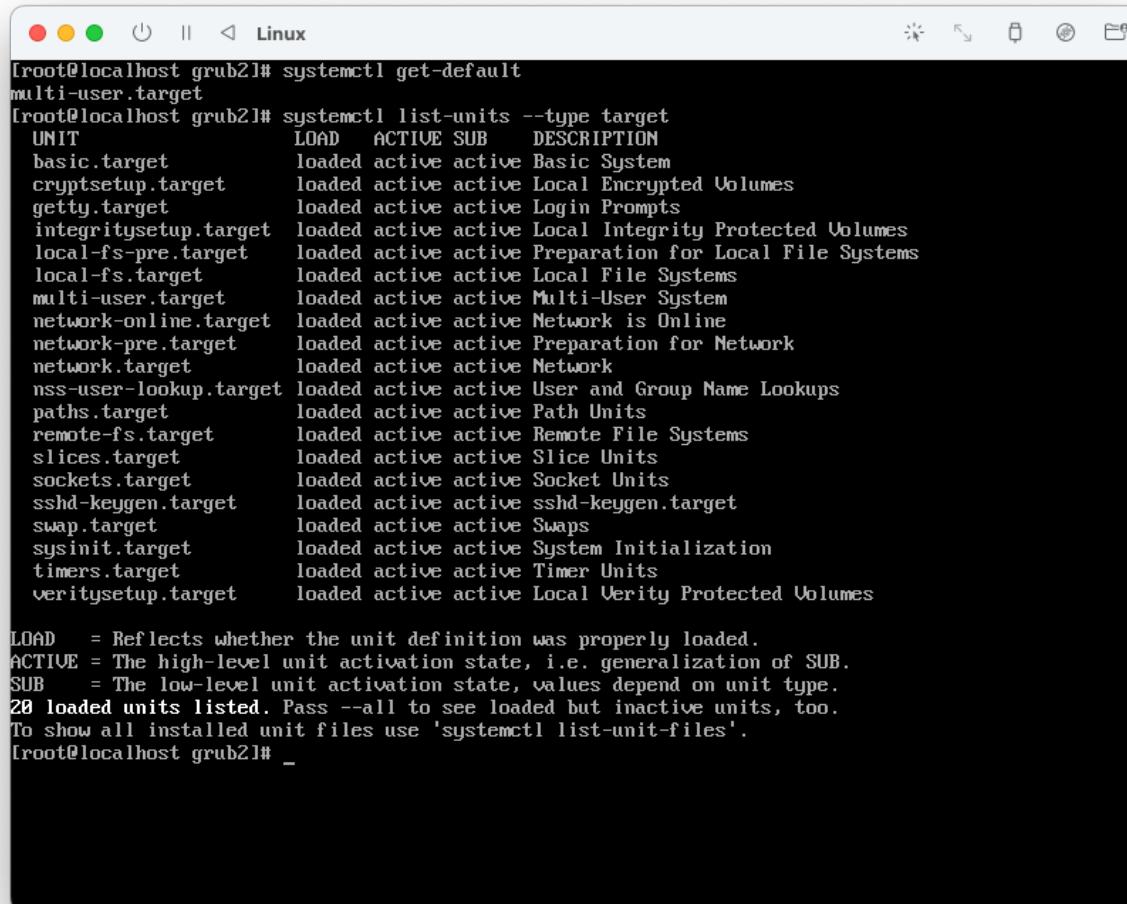
# 7.L Comparación runlevels SysV y targets systemd

| Runlevel | Target Units                        | Description                               |
|----------|-------------------------------------|-------------------------------------------|
| 0        | runlevel0.target, poweroff.target   | Shut down and power off the system.       |
| 1        | runlevel1.target, rescue.target     | Set up a rescue shell.                    |
| 2        | runlevel2.target, multi-user.target | Set up a non-graphical multi-user system. |
| 3        | runlevel3.target, multi-user.target | Set up a non-graphical multi-user system. |
| 4        | runlevel4.target, multi-user.target | Set up a non-graphical multi-user system. |
| 5        | runlevel5.target, graphical.target  | Set up a graphical multi-user system.     |
| 6        | runlevel6.target, reboot.target     | Shut down and reboot the system.          |

- Para ver, cambiar o configurar targets systemd, se emplea el comando `systemctl`.

# 7.L comando systemctl

- Target por defecto: **systemctl get-default** (resuelve el enlace simbólico en /etc/systemd/system/default.target y muestra el resultado)
- Mostrar todos los targets: **systemctl list-units --type target**



```
[root@localhost grub2]# systemctl get-default
multi-user.target
[root@localhost grub2]# systemctl list-units --type target
UNIT LOAD ACTIVE SUB DESCRIPTION
basic.target loaded active active Basic System
cryptsetup.target loaded active active Local Encrypted Volumes
getty.target loaded active active Login Prompts
integritysetup.target loaded active active Local Integrity Protected Volumes
local-fs-pre.target loaded active active Preparation for Local File Systems
local-fs.target loaded active active Local File Systems
multi-user.target loaded active active Multi-User System
network-online.target loaded active active Network is Online
network-pre.target loaded active active Preparation for Network
network.target loaded active active Network
nss-user-lookup.target loaded active active User and Group Name Lookups
paths.target loaded active active Path Units
remote-fs.target loaded active active Remote File Systems
slices.target loaded active active Slice Units
sockets.target loaded active active Socket Units
sshd-keygen.target loaded active active sshd-keygen.target
swap.target loaded active active Swaps
sysinit.target loaded active active System Initialization
timers.target loaded active active Timer Units
veritysetup.target loaded active active Local Verity Protected Volumes

LOAD = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB = The low-level unit activation state, values depend on unit type.
20 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
[root@localhost grub2]# _
```

# 7.L comando systemctl

---

- Cambiar el target por defecto: **systemctl set-default nombre.target**. Este comando reemplaza **/etc/systemd/system/nombre.target** con un enlace simbólico a **/usr/lib/systemd/system/nombre.target**
- Cambiar el target actual: **systemctl isolate nombre.target** (p.e. **systemctl isolate multi-user.target**). Este comando inicia la unidad target llamada “nombre” y todas las unidades que dependen de ella, y detiene todas las demás.
- Cambiar a modo de rescate (single user): **systemctl rescue**

# 7.L Comparación runlevels SysV y targets systemd

---

| Old Command                          | New Command                                       | Description                          |
|--------------------------------------|---------------------------------------------------|--------------------------------------|
| runlevel                             | <code>systemctl list-units --type target</code>   | Lists currently loaded target units. |
| <code>telinit <i>runlevel</i></code> | <code>systemctl isolate <i>name.target</i></code> | Changes the current target.          |

- Los comandos runlevel y telinit siguen existiendo por compatibilidad

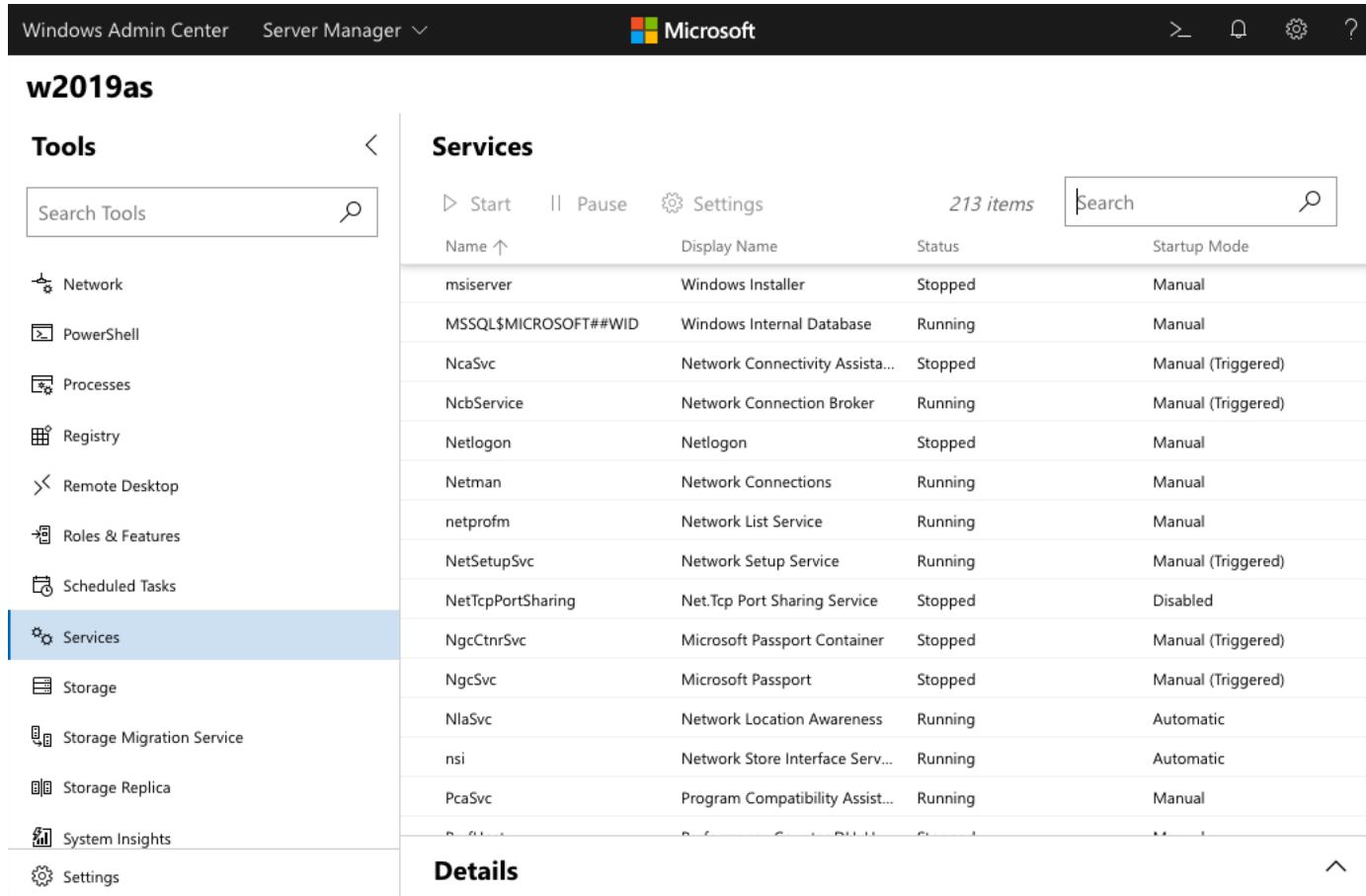
# 7.L Inicio y detención de servicios

The screenshot shows the Cockpit web interface running on a local host. The URL in the browser is <https://192.168.64.2:9090/system/services>. The interface has a dark theme with a sidebar on the left containing navigation links like System, Overview, Logs, Storage, Networking, Accounts, Services (which is selected), Tools, Applications, Diagnostic Reports, Kernel Dump, SELinux, Software Updates, Subscriptions, and Terminal. The main content area is titled "Services" and lists various system services. Each service entry includes the service name, a brief description, its current state (Running or Not running), and its enable status (Enabled or Disabled). The services listed are:

| Service                                                                                                                                                                                      | Description                                 | State       | Enable   |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|-------------|----------|
| auditd                                                                                                                                                                                       | Security Auditing Service                   | Running     | Enabled  |
| blk-availability                                                                                                                                                                             | Availability of block devices               | Not running | Disabled |
| chrony-wait                                                                                                                                                                                  | Wait for chrony to synchronize system clock | Not running | Disabled |
| chronyd                                                                                                                                                                                      | NTP client/server                           | Running     | Enabled  |
| cockpit-motd                                                                                                                                                                                 | Cockpit motd updater service                | Not running | Static   |
| cockpit-wsinstance-http                                                                                                                                                                      | Cockpit Web Service http instance           | Not running | Static   |
| cockpit-wsinstance-<br><a href="https://e3b0c44298fc1c149afb4c8996fb92427ae41e4649b649b934ca495991b7852b855">https://e3b0c44298fc1c149afb4c8996fb92427ae41e4649b649b934ca495991b7852b855</a> | Cockpit Web Service https instance          | Running     |          |
| cockpit                                                                                                                                                                                      | Cockpit Web Service                         | Running     | Static   |
| console-getty                                                                                                                                                                                | Console Getty                               | Not running | Disabled |

- Desde la consola web pueden gestionarse los servicios y los targets systemd

# 7.W Inicio y detención de servicios



The screenshot shows the Windows Admin Center interface for managing services. The left sidebar has a 'Tools' section with various icons and names: Network, PowerShell, Processes, Registry, Remote Desktop, Roles & Features, Scheduled Tasks, Services (which is selected and highlighted in blue), Storage, Storage Migration Service, Storage Replica, System Insights, and Settings. The main right pane is titled 'Services' and displays a table of 213 items. The columns are Name, Display Name, Status, and Startup Mode. The table lists numerous services like msiserver, MSSQL\$MICROSOFT##WID, NcaSvc, NcbService, Netlogon, Netman, netprofm, NetSetupSvc, NetTcpPortSharing, NgcCtnrSvc, NgcSvc, NlaSvc, nsi, and PcaSvc, each with their respective status (e.g., Stopped, Running) and startup mode (e.g., Manual, Manual (Triggered), Automatic). A search bar is at the top of the table.

| Name ↑                | Display Name                    | Status  | Startup Mode       |
|-----------------------|---------------------------------|---------|--------------------|
| msiserver             | Windows Installer               | Stopped | Manual             |
| MSSQL\$MICROSOFT##WID | Windows Internal Database       | Running | Manual             |
| NcaSvc                | Network Connectivity Assista... | Stopped | Manual (Triggered) |
| NcbService            | Network Connection Broker       | Running | Manual (Triggered) |
| Netlogon              | Netlogon                        | Stopped | Manual             |
| Netman                | Network Connections             | Running | Manual             |
| netprofm              | Network List Service            | Running | Manual             |
| NetSetupSvc           | Network Setup Service           | Running | Manual (Triggered) |
| NetTcpPortSharing     | Net.Tcp Port Sharing Service    | Stopped | Disabled           |
| NgcCtnrSvc            | Microsoft Passport Container    | Stopped | Manual (Triggered) |
| NgcSvc                | Microsoft Passport              | Stopped | Manual (Triggered) |
| NlaSvc                | Network Location Awareness      | Running | Automatic          |
| nsi                   | Network Store Interface Serv... | Running | Automatic          |
| PcaSvc                | Program Compatibility Assist... | Running | Manual             |

- En W2019, desde Windows Admin Center, en la opción “Services” se puede indicar qué servicios se arrancan de forma automática o manual (los runlevels/targets no existen en Windows).