

PROMETEO

Unidad 3: Instalación y configuración de sistemas operativos

Antes de instalar un sistema operativo (SO), da un paso atrás y verifica dos cosas: si tu hardware está a la altura y si el software que necesitas convivirá sin fricciones.

Sesión 8 – Requisitos y compatibilidad hardware/software

Antes de instalar un sistema operativo (SO), da un paso atrás y verifica dos cosas: si tu hardware está a la altura y si el software que necesitas convivirá sin fricciones. Saltarte esta verificación previa suele terminar en equipos lentos, cuelgues aleatorios, funciones deshabilitadas (como Wi-Fi o cámara) o, directamente, en que tus aplicaciones críticas no arranquen.

Requisitos de hardware: mínimo vs. recomendado

Los fabricantes publican requisitos mínimos (CPU, RAM, almacenamiento, GPU), pero esos números solo garantizan que el sistema arranque, no que vaya fluido. Para un uso realista, apunta siempre a los requisitos recomendados o algo más si vas a ejecutar cargas exigentes (edición, virtualización, ciencia de datos).

Tres elementos marcan la diferencia:

- **CPU (arquitectura y capacidades)**: no es solo la velocidad (GHz). Mira si es x86_64 o ARM, si soporta virtualización por hardware (Intel VT-x/AMD-V), cifrado, e instrucciones especiales (AVX/AVX2/NEON) que algunas apps exigen. Para Windows 11, además, TPM 2.0 y Secure Boot.
- **RAM**: 4 GB "sirve" para un arranque básico; 8 GB es lo mínimo recomendable para escritorio moderno; 16 GB si harás multitarea real, VMs o IDEs pesados.
- **Almacenamiento**: no es solo capacidad; tipo e IOPS importan. Un SSD NVMe hace que el mismo sistema se sienta "nuevo" frente a un HDD. Calcula un colchón del 30–40 % libre para swap, temporales y actualizaciones.

Compatibilidad de software y controladores (drivers)

El SO necesita "traductores" para hablar con el hardware. Esos traductores son los drivers. En Windows los suele mantener el fabricante; en Linux, el kernel integra soporte para mucho hardware, pero componentes muy nuevos o chipsets específicos (Wi-Fi, GPU) pueden requerir drivers adicionales o no estar soportados aún.

Si tus aplicaciones dependen de aceleración por GPU (NVIDIA CUDA), de OpenGL/DirectX/Vulkan o de periféricos (tabletas, escáneres), verifica versiones y drivers soportados.

Factor firmware: BIOS/UEFI, modo de arranque y particionado

Tu placa usa BIOS o UEFI. UEFI habilita Secure Boot, discos GPT y arranques más rápidos. Algunos SO (o distros) funcionan con Secure Boot activado; otros necesitan desactivarlo o registrar su cargador firmado. Comprueba también si tu almacenamiento está en modo AHCI (recomendado) o RAID/Intel RST, que puede requerir drivers durante la instalación.

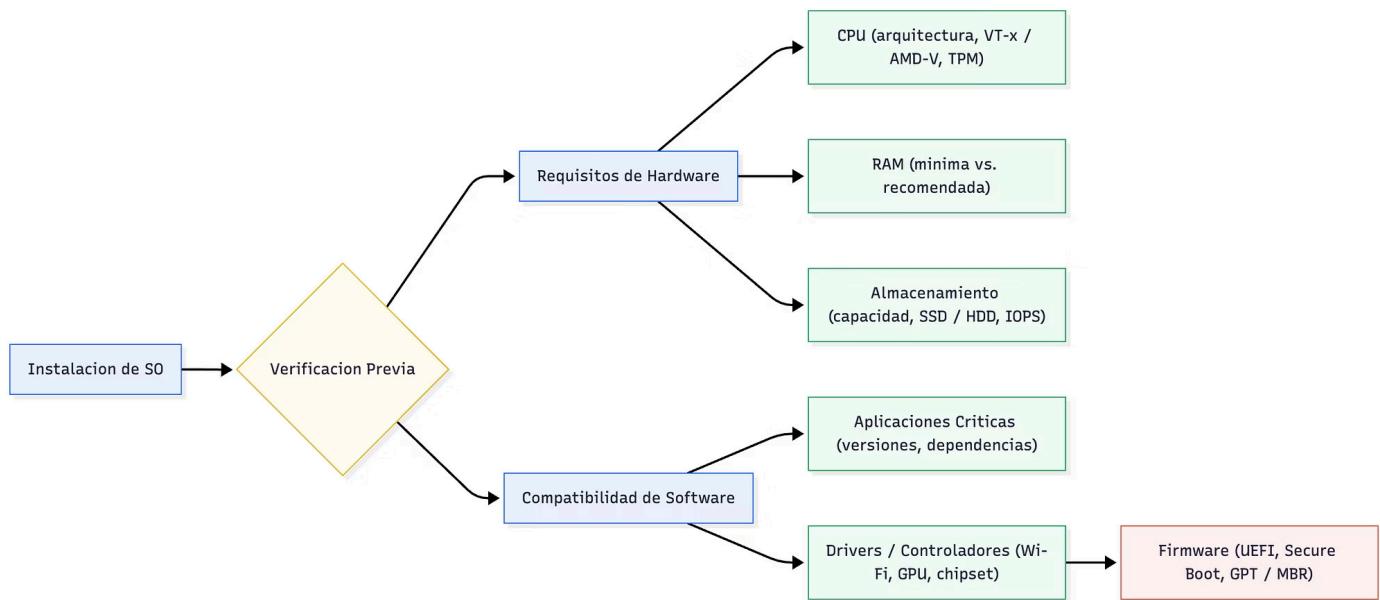
El plan realista

Tu objetivo no es "instalar como sea": es instalar y trabajar sin sorpresas. Por eso, además de leer la ficha técnica del SO, necesitas probar (cuando puedas) con un Live USB o revisar matrices de compatibilidad del fabricante. Una hora de verificación previa te ahorrará días de frustración post-instalación.



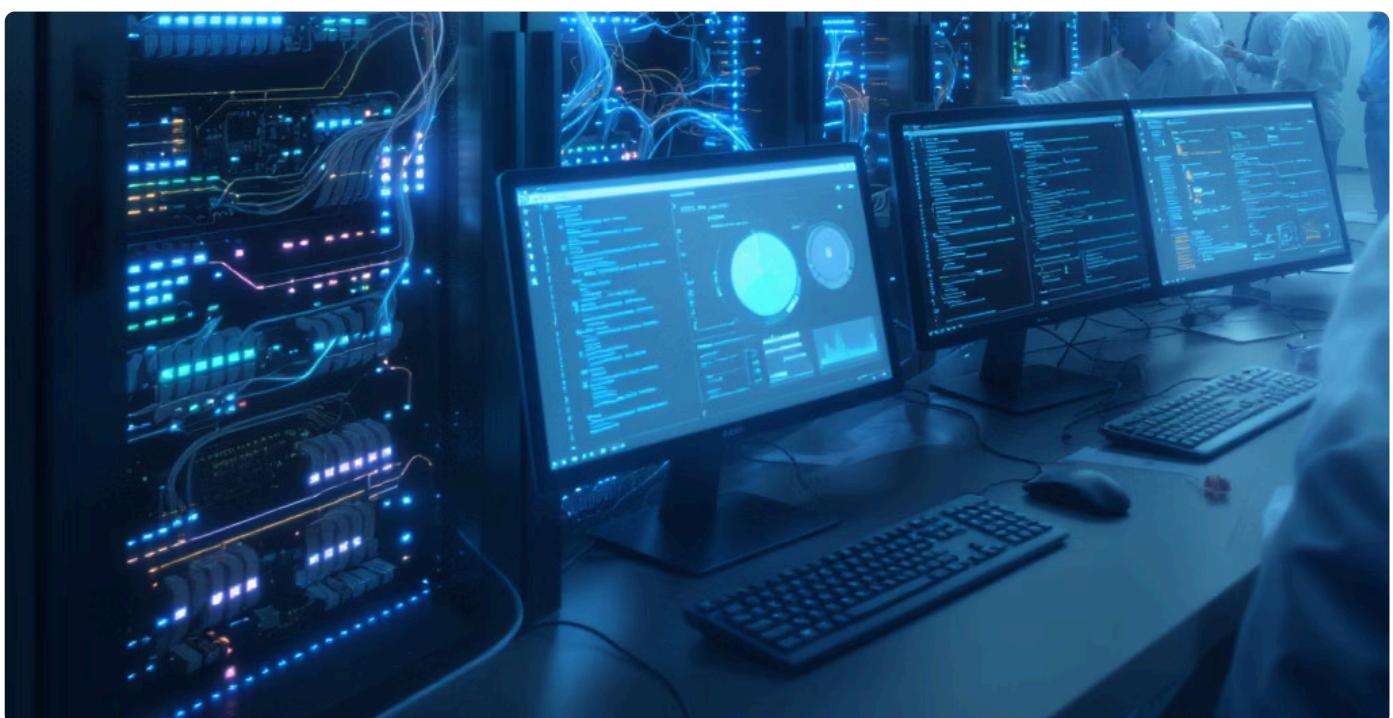
Esquema Visual

Este diagrama sintetiza la verificación previa en dos pilares —hardware y software— y sus subcomponentes clave.



Cómo leerlo:

- **C (Hardware)**: valida arquitectura, capacidades de CPU, RAM y tipo de disco (SSD >> HDD).
- **D (Software)**: comprueba que tus aplicaciones críticas están soportadas en ese SO/version, y que existen drivers estables para tu Wi-Fi/GPU/chipset.
- **J (Firmware)**: UEFI/Secure Boot y esquema de particionado pueden condicionar la instalación.





Caso de Estudio – HPE (Hewlett Packard Enterprise): Certificación y matrices de compatibilidad

Contexto

HPE vende servidores que deben funcionar 24/7 con Windows Server, RHEL, SUSE, VMware ESXi, etc. Un driver inestable o una tarjeta no soportada no es un pequeño inconveniente: es interrupción de negocio.

Estrategia

HPE ejecuta miles de horas de pruebas por combinación de hardware (CPU, RAM, NICs, controladoras RAID, GPUs) y SO/hipervisor. El resultado se publica como Matriz de Compatibilidad: un documento vivo donde puedes verificar, por modelo y versión, si un componente está soportado, qué driver/firmware necesitas y bajo qué condiciones (p. ej., BIOS mínima o microcódigo).

- **Drivers validados y firmados:** se distribuyen mediante Service Packs o repositorios oficiales, reduciendo riesgos de incompatibilidad.
- **Cadena completa:** no solo testean el SO. Validan firmware de controladoras, microcódigo de CPU, BIOS/UEFI y versiones de hipervisores para garantizar estabilidad "de extremo a extremo".

Resultado

Los clientes usan estas matrices antes de comprar o actualizar. Así evitan "sorpresa" y planifican ventanas de mantenimiento con el paquete exacto de BIOS/firmware driver. En entornos críticos, esta gobernanza de compatibilidades es la diferencia entre operar y parar.

Lección trasladable a tu entorno: aunque no administres un data center, imita el procedimiento: verifica soporte oficial, usa drivers del fabricante, alinea BIOS/UEFI y no mezcles versiones al azar.

Herramientas y Consejos (aplicación inmediata)

"Can You RUN It" (PC)

Útil para software y juegos en Windows: analiza tu equipo y compara con requisitos. No sustituye al criterio técnico, pero te da un semáforo rápido sobre CPU/RAM/GPU.

Live USB de Linux (prueba sin instalar)

Arranca Ubuntu, Fedora o una distro ligera desde USB y comprueba Wi-Fi, touchpad, suspensión, audio, brillo, GPU híbrida... Si algo falla en Live, fallará tras instalar.

Comandos útiles: lspci, lsusb, inxi -F, lshw para inventariar hardware.

WSL en Windows (herramientas Linux sin formatear)

Si dependes de Windows (p. ej., Adobe, CAD) pero necesitas terminales Linux, WSL es una solución equilibrada. Revisa si tu flujo requiere systemd, contenedores o acceso GPU (WSLg con soporte gráfico).

Listas de compatibilidad y catálogos "certified"

Ubuntu Certified Hardware, Fedora HCL y las matrices del fabricante (HPE, Dell, Lenovo) son tu primera parada.

Para GPU: revisa NVIDIA (propietario) vs Nouveau (open source); para AMD/Intel, el kernel suele cubrir bien, pero valida versión del kernel y mesa si dependes de Vulkan/OpenCL.

Windows PC Health Check / Requisitos Windows 11

Si vas a Windows 11, confirma TPM 2.0, Secure Boot, CPU soportada y 8 GB+ de RAM para un uso cómodo. Ten en cuenta que forzar instalaciones sin cumplir puede comprometer soporte y actualizaciones.

Almacenamiento: más allá de "tengo espacio"

Si usas HDD, valora migrar a SSD SATA al menos; si ya usas SSD, mejor NVMe.

Planifica particiones: SO, datos, y un colchón para actualizaciones y cache.

En Linux, considera btrfs o ZFS si te interesan snapshots y resiliencia (requieren más RAM y saber gestionarlos).

Firmware y BIOS/UEFI al día

Actualiza BIOS/UEFI antes de instalar si el fabricante corrige compatibilidad de RAM/CPU/NVMe.

Ajustes clave: AHCI para SSD, Desactivar Fast Boot temporalmente durante la instalación, Secure Boot según distribución.

Redes y Wi-Fi "sensibles"

Broadcom/Realtek suelen requerir atención en Linux (drivers propietarios o DKMS); Intel suele ir mejor soportado.

Lleva siempre un dongle USB Ethernet en el kit de instalación por si el Wi-Fi no funciona durante el proceso.

Plan B: distros ligeras y entornos alternativos

Para portátiles antiguos: Lubuntu/Xubuntu, Linux Mint Xfce o Fedora LXQt. Cambiar de GNOME/KDE a Xfce/LXQt reduce RAM y CPU.

Checklist exprés antes de instalar



CPU y arquitectura → OK



RAM ≥ recomendada → OK



SSD con ≥ 30–40 % libre → OK



UEFI/Secure Boot/TPM → OK (si aplica)



Drivers GPU/Wi-Fi identificados → OK



Apps críticas soportadas (versiones/SDK) → OK



Plan de recuperación (backup) → OK

Mitos y Realidades

X Mito: "Si mi PC cumple los requisitos mínimos, el SO funcionará perfectamente."

FALSO. Los requisitos mínimos solo garantizan el arranque del sistema operativo, no una experiencia de usuario fluida y eficiente. Con especificaciones básicas como 4 GB de RAM y un disco duro HDD, realizar tareas cotidianas como abrir múltiples pestañas del navegador o ejecutar un entorno de desarrollo integrado (IDE) puede generar un rendimiento extremadamente lento y cuellos de botella constantes.

Realidad: Para asegurar un uso diario sin fricciones y una experiencia satisfactoria, es crucial apuntar a los requisitos recomendados o superiores. Considera mejorar tu equipo añadiendo una unidad de estado sólido (SSD) para mayor velocidad y más memoria RAM para una mejor multitarea.

X Mito: "En Linux, todo el hardware funciona automáticamente."

FALSO. Aunque el soporte de hardware en Linux ha mejorado drásticamente a lo largo de los años, aún existen casos en los que no todos los componentes funcionan de manera plug-and-play. Dispositivos como tarjetas Wi-Fi específicas (ej. Broadcom o algunas Realtek) y tarjetas gráficas (especialmente NVIDIA) pueden requerir la instalación de drivers propietarios, la actualización del kernel o ajustes específicos (como la gestión de Secure Boot y firmas de módulos).

Realidad: Siempre es recomendable probar la compatibilidad de tu hardware utilizando una distribución Live USB antes de la instalación definitiva. Consulta la Lista de Compatibilidad de Hardware (HCL) de la distribución que planeas usar y ten en cuenta que podrías necesitar pasos adicionales para configurar ciertos periféricos.

❑ Resumen Final (para examen)

- Verifica hardware y software antes de instalar: CPU/arquitectura, RAM, SSD, UEFI/Secure Boot/TPM y drivers.
- Apunta a requisitos recomendados (no mínimos) para un uso fluido.
- Drivers y firmware son críticos, especialmente en Linux (GPU/Wi-Fi) y en entornos con Secure Boot.
- Usa Live USB, matrices de compatibilidad y herramientas como WSL para minimizar riesgos y asegurar productividad desde el primer arranque.

Sesión 9 – Preparación de hardware, BIOS/UEFI, arranque seguro

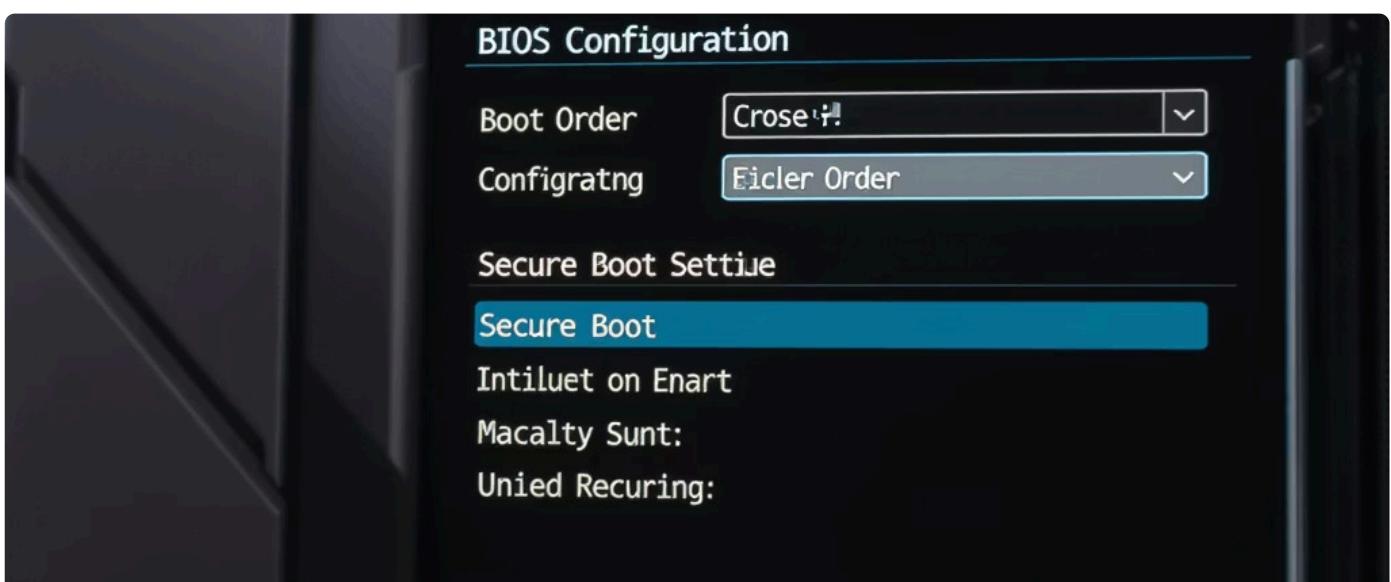
Antes de instalar cualquier sistema operativo, hay un paso que nunca debes saltarte: preparar el hardware y el firmware que lo gobierna al encender el equipo. Ese firmware puede ser la BIOS (Basic Input/Output System) de toda la vida o su sucesor, UEFI (Unified Extensible Firmware Interface). Su función es inicializar los componentes (CPU, memoria, almacenamiento, gráficos, red), realizar comprobaciones básicas (POST) y, después, entregar el control al cargador de arranque del sistema operativo en el dispositivo que elijas.

¿Qué hace tan estratégica esta fase?

Aquí decides tres cosas que condicionan toda la instalación:

1. **De dónde arranca el equipo (Boot Order)**. Si vas a instalar desde un USB, debe estar por delante del disco interno en el orden de arranque o usar un menú de "one-time boot".
2. **Nivel de seguridad del arranque (Secure Boot)**. UEFI incorpora un mecanismo de confianza que permite arrancar únicamente binarios firmados. Protege contra malware de bajo nivel (bootkits) que intenta cargarse antes del sistema operativo.
3. **Parámetros de bajo nivel** que optimizan compatibilidad y rendimiento: virtualización (Intel VT-x/AMD-V y VT-d/IOMMU), modo de almacenamiento (AHCI/RAID), soporte para TPM, C-States, perfiles de energía, o la activación del CSM/Legacy si, por motivos de compatibilidad, debieras arrancar como si fuese BIOS tradicional.

En el contexto actual, UEFI aporta ventajas claras: interfaz gráfica y navegación con ratón, soporte para GPT (particiones modernas, discos >2 TB), arranque más rápido y mejores capas de seguridad (Secure Boot, medido por TPM). Además, la relación entre UEFI y el sistema operativo es más estrecha: Windows 11, por ejemplo, exige UEFI, Secure Boot y TPM activos para su instalación en equipos soportados.



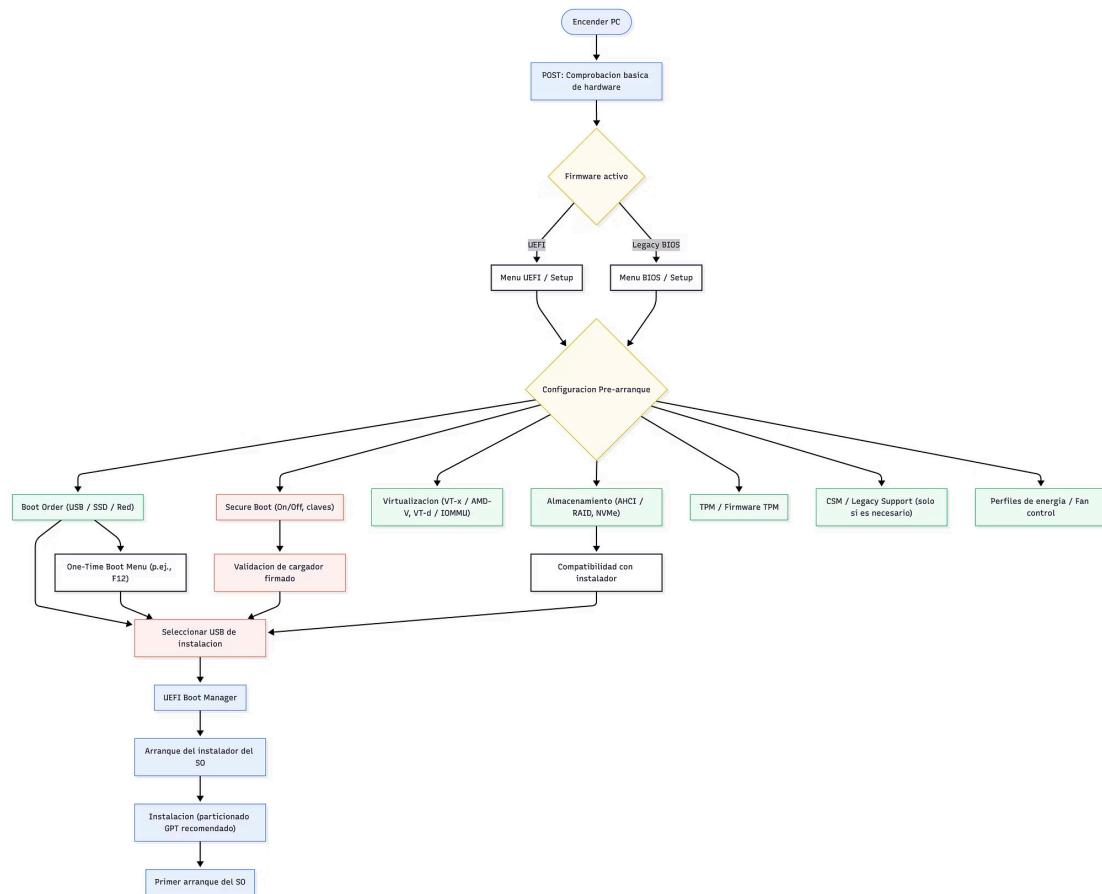
A nivel práctico, la preparación del hardware implica:

- Actualizar el firmware (BIOS/UEFI) para corregir fallos, ampliar compatibilidad de memoria/CPU/SSD y reforzar la seguridad.
- Comprobar almacenamiento y memoria con diagnósticos integrados antes de instalar para evitar errores posteriores.
- Definir el esquema de particionado adecuado al modo de arranque: UEFI ➔ GPT; BIOS/Legacy ➔ MBR (aunque hoy lo recomendable es UEFI+GPT).
- Planificar escenarios especiales, como dual-boot con Windows (coherencia entre UEFI/Legacy, comprobación de BitLocker, desactivar "Fast Startup" en Windows para evitar bloqueos de particiones NTFS), o el uso de cifrado a nivel de disco (TPM + BitLocker/LUKS).

Piensa en la BIOS/UEFI como la torre de control del aeropuerto: si no autoriza el orden correcto de despegues (boot order), si el control de seguridad (Secure Boot) no valida credenciales, o si el plan de vuelo (parámetros de hardware) no encaja con el modelo de avión (SO), el aterrizaje seguro del sistema operativo será difícil. Preparar aquí te evita reinstalaciones innecesarias y diagnóstico a destiempo.

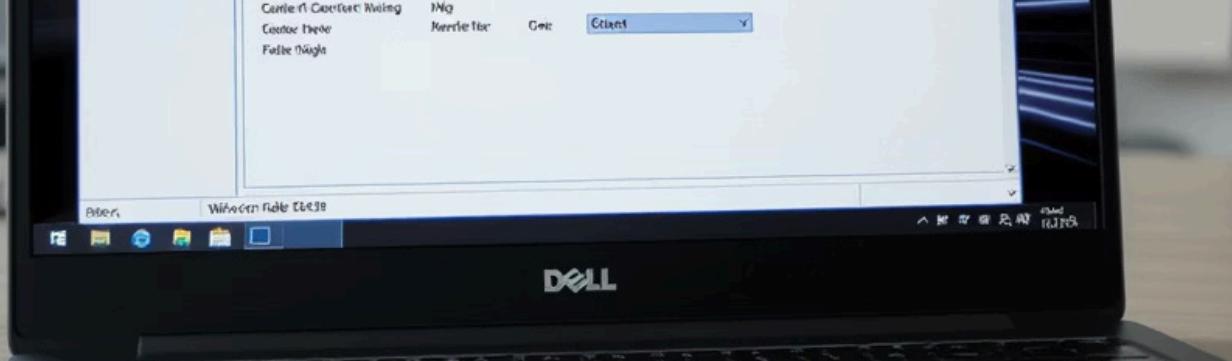


Esquema Visual



Cómo interpretarlo

- **POST** verifica que el hardware básico responde (RAM, CPU, GPU, teclado).
- **UEFI/BIOS Setup**: aquí ajustas todo lo relevante antes del sistema operativo.
- **Boot Order** determina el dispositivo desde el que se intentará arrancar; One-Time Boot te deja elegir temporalmente (muy útil para instalar desde USB sin cambiar la lista permanente).
- **Secure Boot** valida firmas del cargador de arranque. Si instalas una distro Linux principal (Ubuntu, Fedora...), suele funcionar activado; en casos especiales puedes desactivarlo temporalmente.
- **Virtualización** permite usar hipervisores (VirtualBox, KVM, Hyper-V) o ejecutar WSL2 con aceleración.
- **Modo de almacenamiento (AHCI/RAID/NVMe)** afecta a rendimiento y a la detección de discos por el instalador.
- **TPM** habilita funciones de seguridad como BitLocker o medición de arranque.
- **CSM/Legacy** solo si el medio no es compatible con UEFI; lo recomendable es UEFI nativo con GPT.



Caso de Estudio – Dell Technologies Contexto

Dell personaliza su firmware UEFI en portátiles y sobremesas empresariales (Latitude, OptiPlex, Precision) y de consumo (Inspiron, XPS). Su objetivo: combinar seguridad, diagnóstico y gestión remota para TI.

Estrategia

- **Acceso y menús claros:** la tecla F2 abre el Setup (UEFI), F12 abre el One-Time Boot Menu. Esto facilita arrancar desde USB en instalaciones, sin cambiar el orden de arranque global.
- **Seguridad reforzada:** opciones visibles para Secure Boot, TPM, contraseñas de supervisor/usuario y configuración de BIOS passwords. Los perfiles permiten bloquear el arranque desde dispositivos extraíbles en entornos corporativos.
- **Compatibilidad y rendimiento:** selección de AHCI/RAID para almacenamiento, soporte NVMe y control de ventiladores/perfil térmico. Esto permite ajustar a escenarios con SSD NVMe de alto rendimiento o arreglar incompatibilidades con instaladores antiguos.
- **Diagnóstico integrado (ePSA):** desde F12, ejecutas pruebas de memoria RAM, disco y gráficos antes de arrancar el sistema operativo. Ideal para descartar hardware defectuoso antes de culpar al instalador.
- **Actualización de firmware simplificada:** Dell permite flashear BIOS/UEFI desde el propio entorno UEFI o desde Windows/Linux con utilidades oficiales. Esto reduce el riesgo y anima a mantener el firmware al día (parches de seguridad, microcódigos de CPU, compatibilidad con nuevos SSD/DDR).

Resultado

Empresas y usuarios técnicos reducen tiempos de soporte: si un instalador no ve el SSD, cambian AHCI/RAID o actualizan firmware; si el USB no arranca, usan F12 o corrigen Secure Boot; si hay dudas de hardware, pasan ePSA y documentan el código de error. La combinación de menús coherentes + diagnóstico + actualizaciones hace que la fase de preparación sea repetible y segura a escala.

Herramientas y Consejos



Acceso rápido al firmware

Prueba F2, F10, F12 o Supr (Del) nada más encender. En Dell, F2 (Setup) y F12 (Boot Menu) son habituales; en HP, Esc/F10; en Lenovo, F1/F2/Enter.

Si usas equipos modernos con "Fast Boot", puede saltarse el tiempo de pulsar teclas; desde el sistema operativo, usa el reinicio avanzado (Windows: Configuración → Recuperación → Inicio avanzado) para entrar en UEFI.



Prepara bien el USB de instalación

Crea el USB con Rufus o Ventoy. Para UEFI nativo, formatea en GPT y sistema de archivos FAT32 (permite Secure Boot y compatibilidad amplia).

Verifica la checksum (SHA256) de la ISO para evitar medios corruptos.



Ajusta UEFI para tu escenario

UEFI + Secure Boot + GPT: configuración recomendada y compatible con Windows 10/11 y la mayoría de distros Linux principales.

Virtualización: activa VT-x/AMD-V y VT-d/IOMMU si vas a usar hipervisores o pasarela de dispositivos.

Almacenamiento: usa AHCI salvo que necesites RAID por hardware/Intel RST. Si cambias de RAID a AHCI en un Windows ya instalado, prepara el cambio (puede requerir ajuste de drivers/registro).



Planifica el dual-boot (Windows + Linux)

Mantén ambos en UEFI con GPT. Evita mezclar UEFI con Legacy.

En Windows, desactiva Fast Startup (inicio rápido) para que no "hiberne" las particiones NTFS.

Si usas BitLocker, suspende temporalmente la protección antes de tocar el gestor de arranque.

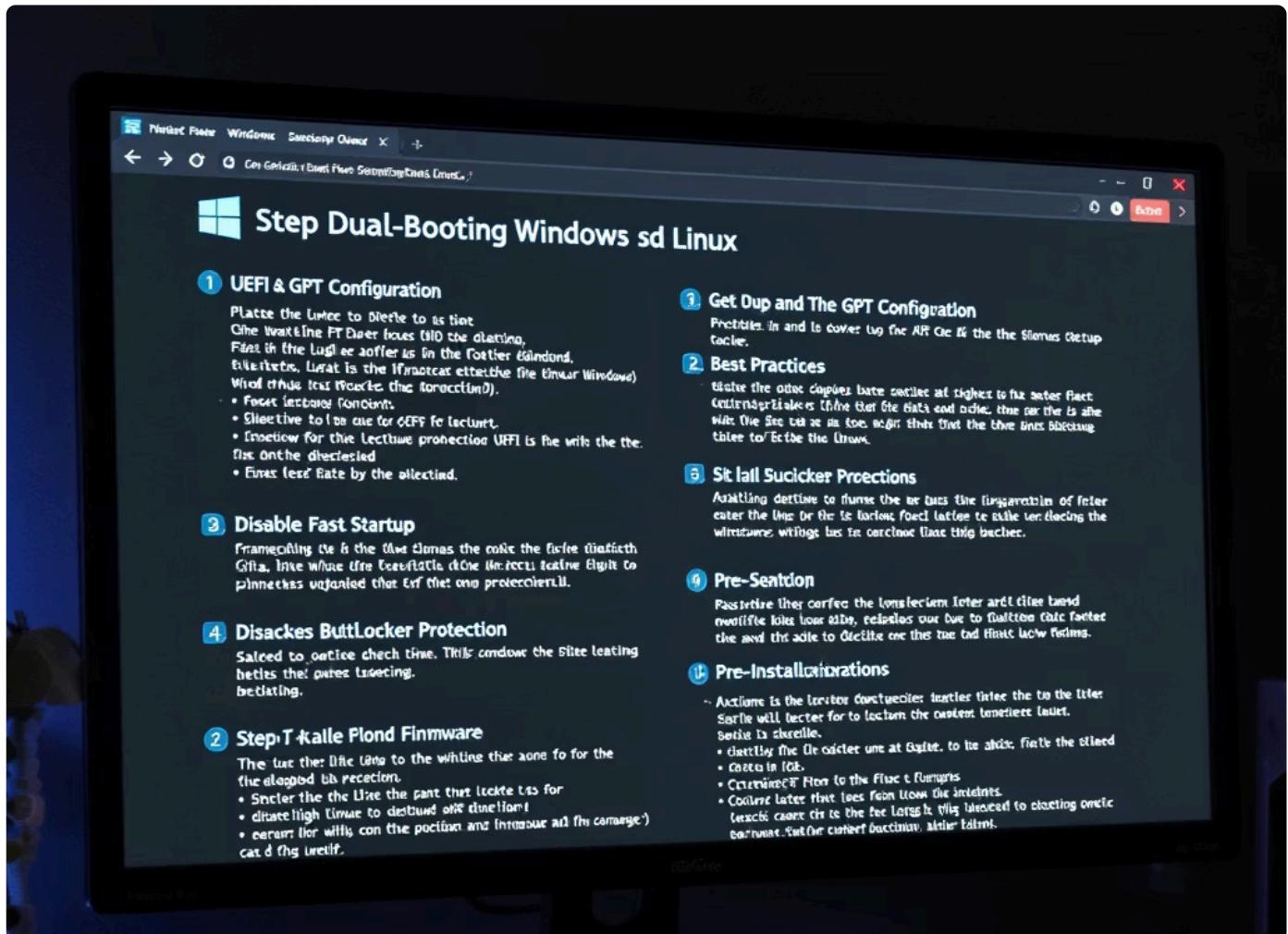
Actualiza con cabeza el firmware

Descarga BIOS/UEFI solo del fabricante, con batería cargada/UPS conectado y sin interrumpir el proceso.

Tras actualizar, revisa que Secure Boot y Boot Order sigan como los dejaste (algunas actualizaciones restablecen valores).

Diagnóstico previo

Si el instalador falla, pasa diagnósticos de RAM y SSD (ePSA en Dell u otras utilidades). Un módulo de memoria defectuoso puede parecer "problema de ISO" y no lo es.



Mitos y Realidades

X Mito: "BIOS y UEFI son lo mismo."

FALSO. UEFI es el sucesor moderno con interfaz más rica, soporte de GPT (discos >2 TB), mejor gestión de arranque y seguridad (Secure Boot). La BIOS tradicional trabaja con MBR y carece de estas capacidades avanzadas.

X Mito: "Secure Boot es un invento de Microsoft para bloquear Linux."

FALSO. Secure Boot es un estándar de la industria para evitar que se cargue malware antes del sistema operativo. Las distribuciones Linux principales firman sus cargadores para funcionar con Secure Boot activado. Solo algunas distros muy personalizadas pueden requerir desactivarlo temporalmente.

□ Resumen Final

- UEFI reemplaza a la BIOS: más seguridad, GPT y mejor compatibilidad.
- Configura Boot Order y usa el One-Time Boot para instalar desde USB.
- Secure Boot protege del malware de arranque; compatible con las principales distros.
- Activa virtualización, ajusta AHCI/RAID y TPM según tu caso.
- Actualiza el firmware con buena práctica y usa diagnósticos antes de instalar.



Sesión 10 – Creación de medios de instalación: USB, PXE, cloud-init

Una instalación exitosa de un sistema operativo no empieza cuando ves la pantalla del instalador, sino mucho antes, al preparar el medio de instalación. Este medio es lo que permite que el ordenador arranque un entorno temporal —normalmente desde un USB, una red PXE, o un entorno cloud-init en la nube— desde el cual el sistema operativo puede copiarse, configurarse y comenzar su vida útil.

El USB de arranque: el estándar actual

Hoy en día, el método más habitual para usuarios y técnicos es el USB de arranque (bootable USB). Se crea a partir de una imagen ISO, que contiene todos los archivos necesarios para la instalación del sistema operativo y un sector de arranque (bootloader). A diferencia de copiar un archivo ISO directamente al pendrive, una herramienta especializada (como Rufus, Ventoy o balenaEtcher) escribe la ISO de forma estructurada, añadiendo ese sector de arranque que la BIOS/UEFI necesita para ejecutarla.

El proceso suele seguir tres pasos:

1. Descargar la imagen ISO oficial (por ejemplo, Ubuntu, Windows o Fedora).
2. Verificar su integridad mediante un checksum SHA256, para asegurarte de que no se ha corrompido.
3. Usar una herramienta de creación de USBs de arranque para grabarla en un pendrive de al menos 8 GB.

Una vez hecho, basta con conectar el USB, entrar en la BIOS/UEFI y establecer el USB como primer dispositivo de arranque.



PXE: la instalación por red

En entornos profesionales, el USB resulta insuficiente. Imagina una empresa con 200 portátiles nuevos que deben recibir la misma imagen del sistema operativo. Conectar un pendrive a cada uno sería inviable. Ahí entra PXE (Preboot Execution Environment): un estándar que permite arrancar ordenadores a través de la red.

Cuando un equipo compatible arranca con PXE habilitado, contacta con un servidor DHCP y un servidor TFTP, descarga un pequeño cargador de arranque (como iPXE o GRUB) y desde ahí obtiene la imagen de instalación del sistema operativo o una imagen corporativa preconfigurada.

Ventajas:

- Automatiza instalaciones masivas.
- Centraliza imágenes y configuraciones.
- Reduce errores humanos.

PXE es la herramienta base de soluciones de despliegue empresarial como Windows Deployment Services (WDS), FOG Project o MAAS (Metal-as-a-Service) de Canonical.

cloud-init: instalación automatizada en la nube

En el ámbito cloud, no existen ni pendrives ni redes locales visibles. Las máquinas virtuales (VM) o instancias se despliegan automáticamente y deben configurarse en su primer arranque. Ahí entra cloud-init, un sistema de inicialización que ejecuta scripts o configuraciones (YAML) en el primer inicio de la instancia.

Permite definir usuarios, contraseñas, claves SSH, paquetes a instalar y scripts de configuración. Por ejemplo, puedes lanzar una VM en AWS EC2 o Azure y, mediante cloud-init, lograr que:

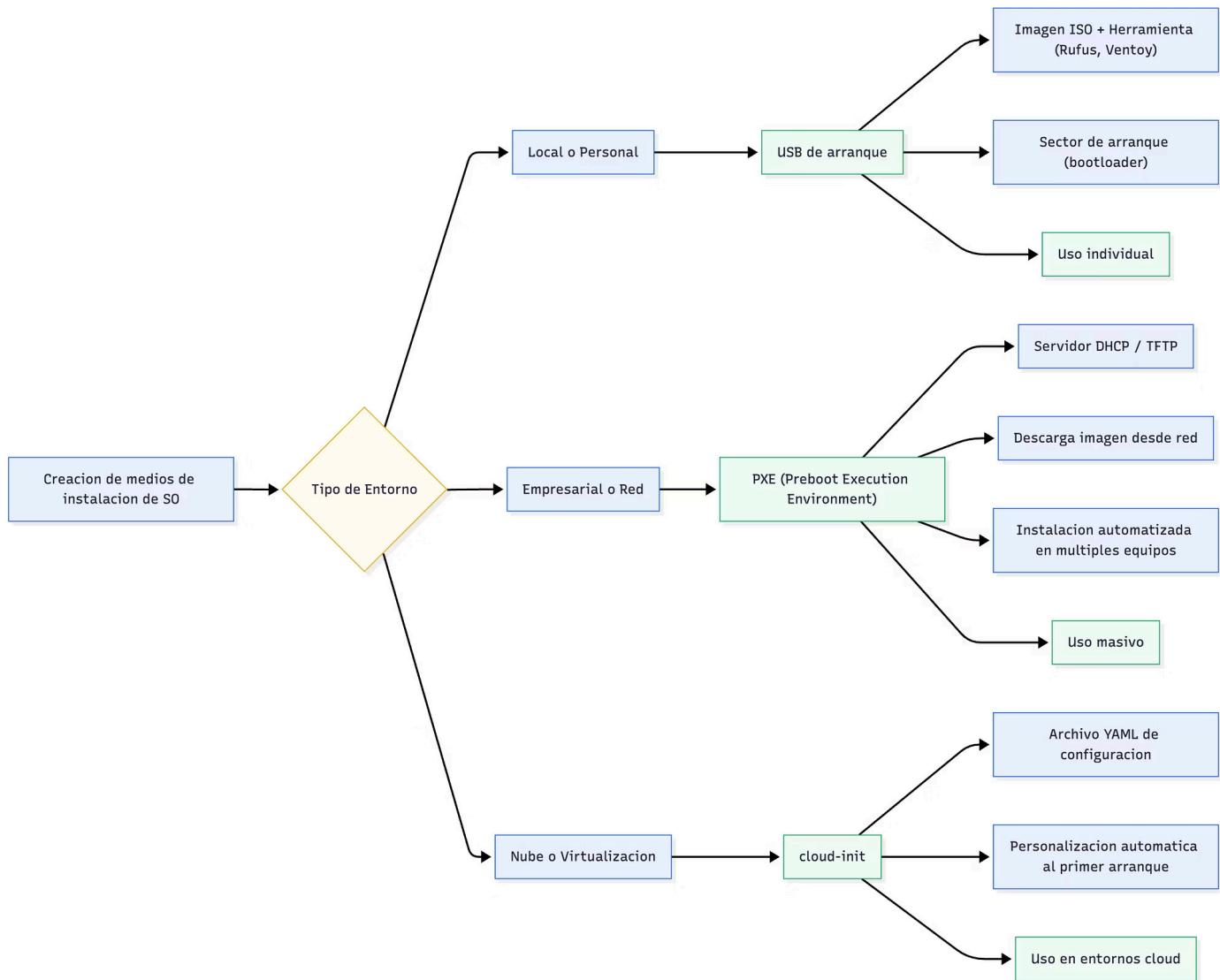
- Cree un usuario administrador.
- Actualice el sistema.
- Monte volúmenes.
- Instale dependencias.

cloud-init es esencial en DevOps y automatización de infraestructura, donde cada segundo cuenta y la coherencia de los despliegues debe ser absoluta.

En resumen, cada entorno tiene su método:

- **USB** → Individual o doméstico.
- **PXE** → Corporativo o masivo.
- **cloud-init** → Virtualización y nube.

Esquema Visual

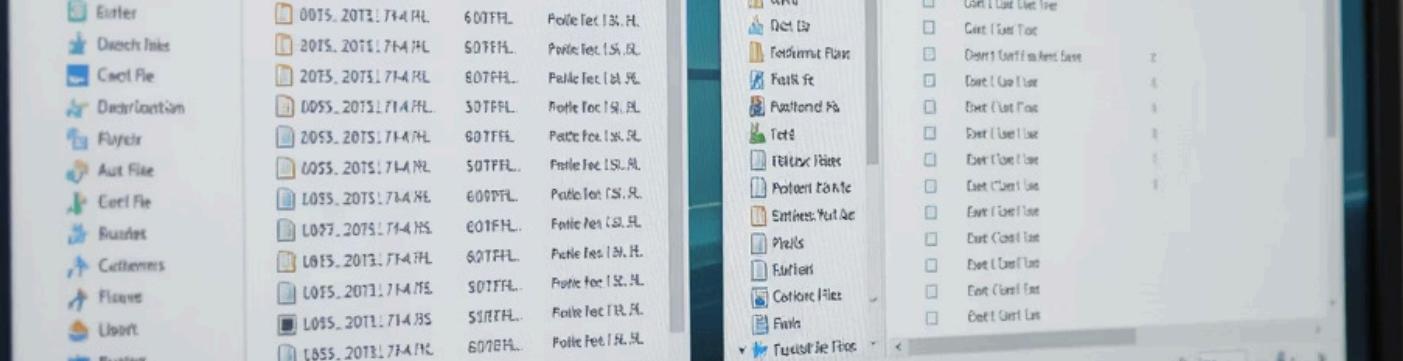


Descripción del esquema

El diagrama representa los tres caminos según el entorno:

- En local, el usuario crea un USB bootable a partir de una ISO mediante herramientas como Rufus.
- En empresas, la instalación se lanza por red usando PXE, que combina servidores DHCP/TFTP con imágenes de instalación centralizadas.
- En la nube, las instancias usan cloud-init para configurarse automáticamente al arrancar por primera vez.

En conjunto, los tres métodos forman la cadena de despliegue moderna que cubre desde el ordenador doméstico hasta el centro de datos y los entornos virtualizados globales.



Caso de Estudio – Rufus y Ventoy

Contexto

Rufus y Ventoy son dos herramientas que han revolucionado la creación de medios de instalación. Antes, crear un USB arrancable requería comandos complejos o utilidades poco intuitivas.

Estrategia de Rufus

Rufus simplificó el proceso al máximo: seleccionas la imagen ISO, eliges el dispositivo USB y presionas Start. Permite seleccionar esquema de particionado (MBR o GPT) y tipo de firmware (BIOS/UEFI). Su ventaja es la rapidez y la compatibilidad casi universal con Windows y Linux.

Estrategia de Ventoy

Ventoy, en cambio, reinventó el concepto. Solo necesitas preparar el USB una vez. Luego puedes copiar directamente varias ISOs al pendrive (Windows, Ubuntu, Fedora, herramientas de rescate...). Al arrancar, Ventoy muestra un menú interactivo que te permite elegir cuál iniciar.

Resultado

- Rufus domina el entorno doméstico y técnico de soporte, ideal para una instalación rápida y segura.
- Ventoy brilla entre técnicos y administradores que necesitan múltiples sistemas en un solo USB, reduciendo tiempo y esfuerzo.

Ambas herramientas son open source y ampliamente utilizadas por profesionales de TI, integrándose incluso en flujos de instalación de fabricantes o en procesos de mantenimiento en campo.

Herramientas y Consejos

Rufus

- Ideal para instalaciones individuales.
- Permite elegir UEFI/BIOS y sistema de archivos (FAT32 o NTFS).
- Disponible en Windows, sin instalación.
- Incluye opciones avanzadas como creación de particiones persistentes en Linux Live.

Ventoy

- Ideal para técnicos o docentes.
- Solo se configura una vez; después basta con copiar ISOs.
- Soporta más de 200 distribuciones de Linux y versiones de Windows.
- Compatible con UEFI, Secure Boot y GPT.

Verificación de imágenes ISO

- Antes de grabar, verifica el hash SHA256 o MD5 del archivo ISO con la web oficial del sistema operativo.
- Esto evita instalaciones fallidas por archivos corruptos o modificados.

PXE Boot

- Usa herramientas como FOG Project (open source) o MAAS (Canonical) para despliegues de red.
- Asegúrate de que tu tarjeta de red soporte arranque PXE y esté activado en la BIOS/UEFI.

cloud-init en la nube

- Define tus configuraciones en formato YAML.
- Incluye comandos runcmd o package_update: true para automatizar tareas iniciales.
- Compatible con AWS, Azure, Google Cloud, OpenStack.

Buenas prácticas generales

- Usa USBs de buena calidad (mínimo 3.0).
- Siempre desmonta el dispositivo de forma segura antes de retirarlo.
- Documenta las configuraciones PXE o scripts cloud-init para futuras automatizaciones.

Mitos y Realidades

X Mito: "Crear un USB de arranque es solo copiar la ISO al pendrive."

FALSO. Una ISO no es un archivo común; contiene sectores de arranque. Se necesita una herramienta especializada que escriba la estructura del disco y el cargador para que la BIOS/UEFI pueda arrancarlo.

X Mito: "PXE es una tecnología antigua y obsoleta."

FALSO. Aunque tiene décadas, PXE sigue siendo esencial en despliegues masivos. Se integra en herramientas modernas de gestión de infraestructura como MAAS o WDS, y en entornos cloud privados donde la automatización del arranque sigue siendo clave.

☐ Resumen Final

- Los medios de instalación son la base del despliegue de cualquier sistema operativo.
- USB de arranque: ideal para uso personal, requiere herramientas como Rufus o Ventoy.
- PXE: permite instalaciones por red a gran escala, usado en empresas y centros de datos.
- cloud-init: automatiza la configuración de máquinas virtuales en la nube.
- Verifica siempre tus ISOs y configura correctamente el orden de arranque en BIOS/UEFI.



Sesión 11 – Particionado, sistemas de archivos y configuración inicial

Nada influye tanto en la estabilidad y el rendimiento de un sistema operativo como cómo organizas el disco antes de la instalación. El proceso arranca con el particionado (dividir lógicamente un disco físico en secciones independientes), sigue con el formateo de cada partición con un sistema de archivos (NTFS, EXT4, APFS, etc.), y culmina con la configuración inicial (usuario, idioma, zona horaria, opciones regionales y de privacidad). Si lo haces bien, ganarás en rendimiento, seguridad y facilidad de mantenimiento; si lo haces mal, te complicarás la vida con cuellos de botella, errores y reinstalaciones innecesarias.

Lo primero es comprender el binomio **esquema de particionado + modo de arranque**:

- **UEFI + GPT (recomendado)**: exige una EFI System Partition (ESP) de ~100–300 MB en FAT32 para almacenar los cargadores de arranque. Permite discos >2 TB y más de cuatro particiones primarias sin trucos.
- **BIOS/Legacy + MBR**: legado histórico. Limitado a 2 TB y 4 particiones primarias (o una extendida con lógicas). Hoy lo usarás solo por compatibilidad.

Después, defines qué particiones tendrá tu instalación. El mínimo funcional es una partición del sistema y, en UEFI, la ESP. Pero en entornos profesionales o cuando esperas crecer, conviene separar responsabilidades:

- **/ (raíz)**: donde vive el SO y la mayoría de ficheros en Linux.
- **/home**: datos personales y configuraciones de usuarios en Linux.
- **/var (opcional, servidores)**: registros, colas, bases de datos.
- **swap**: memoria de intercambio; útil para hibernación o contingencias.
- **/boot (opcional)**: arranque en setups cifrados o con ciertos filesystems.
- **ESP (UEFI)**: imprescindible en UEFI para GRUB/systemd-boot/Windows Boot Manager.

En Windows, típicamente verás la partición del sistema (NTFS) y una reservada con metadatos de arranque. En macOS, APFS se organiza en contenedores con volúmenes lógicos dinámicos.

La siguiente decisión crítica es el **sistema de archivos (filesystem)**, que determina cómo se organizan y se accede a los datos:

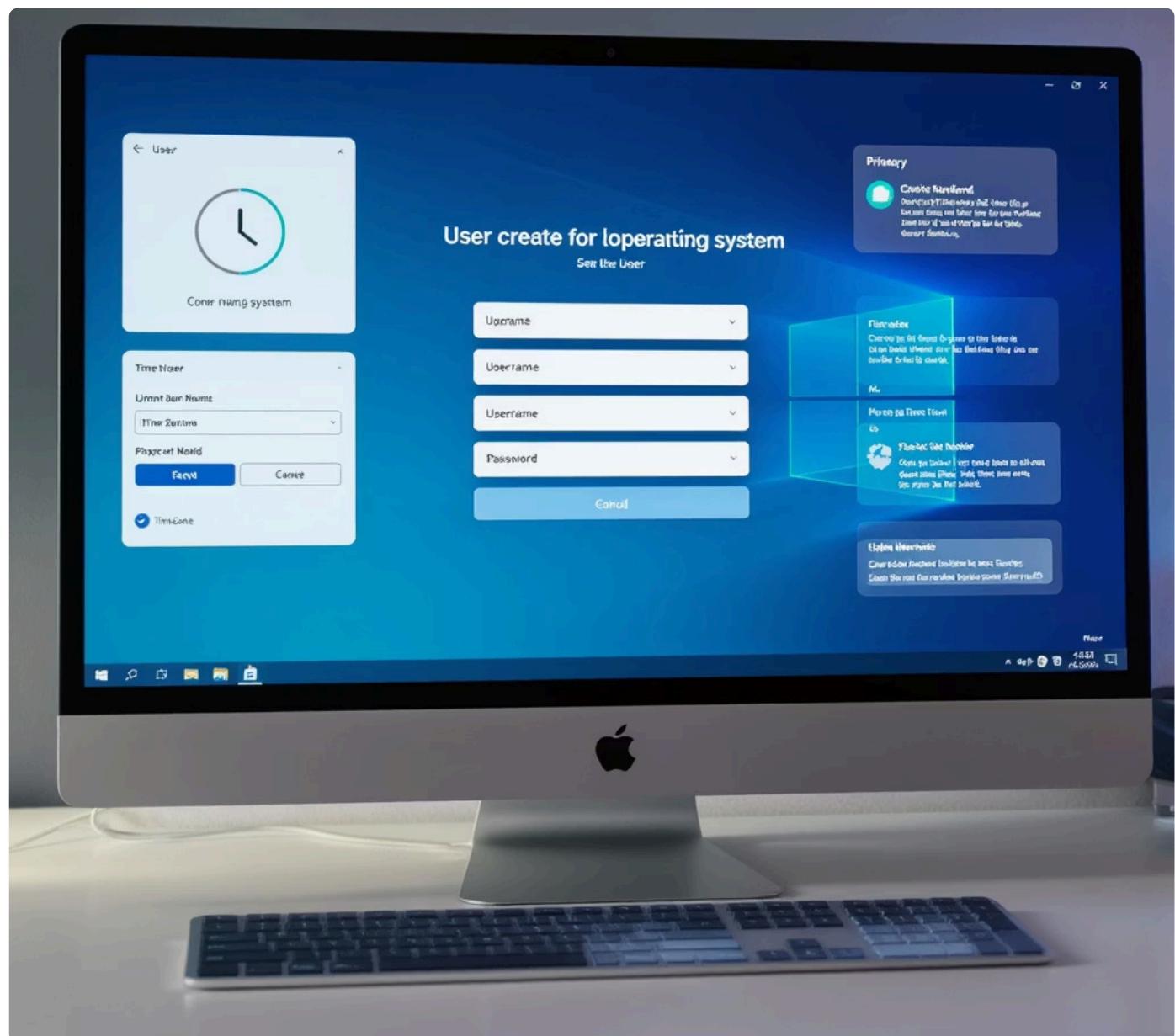
- **Windows**: NTFS (nativo, permisos, compresión, cifrado EFS).
- **Linux**: EXT4 (equilibrado y maduro), XFS (gran rendimiento en ficheros grandes), Btrfs/ZFS (snapshots, compresión y checksums para integridad).
- **macOS**: APFS (snapshots, volúmenes lógicos, cifrado integrado).

No es solo "qué funciona"; es qué te conviene según el uso. Por ejemplo, si vas a gestionar máquinas virtuales pesadas, XFS o EXT4 con opciones ajustadas pueden rendir mejor; si te preocupa la integridad de datos y la recuperación, ZFS/Btrfs con snapshots y checksum te aportan tranquilidad (a cambio de aprender su administración).

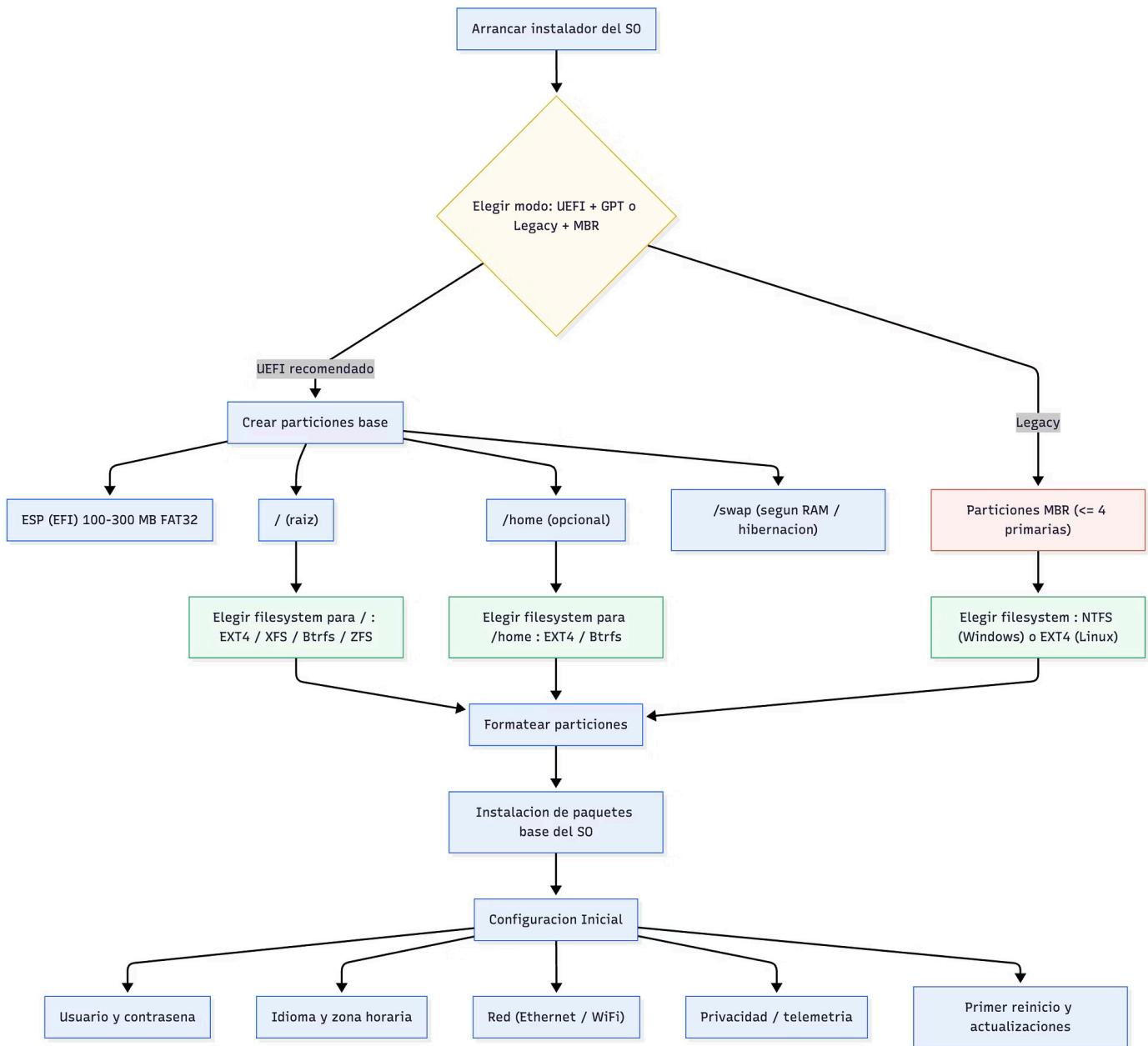
Finalmente, la **configuración inicial** cementa la seguridad y la experiencia:

- Creación del usuario y contraseña (evita usar "admin" para el día a día).
- Zona horaria e idioma (impactan en logs, paquetes, teclado).
- Privacidad (telemetría, servicios opcionales).
- Red básica (Ethernet/WiFi) para poder actualizar el sistema en cuanto arranque.

Una buena metáfora: el disco es un edificio; las particiones son los pisos, el sistema de archivos es la distribución y el cableado, y la configuración inicial es entregar las llaves, definir normas y encender los servicios. Si diseñas el edificio para crecer, no tendrás que derribarlo cuando quieras ampliar.

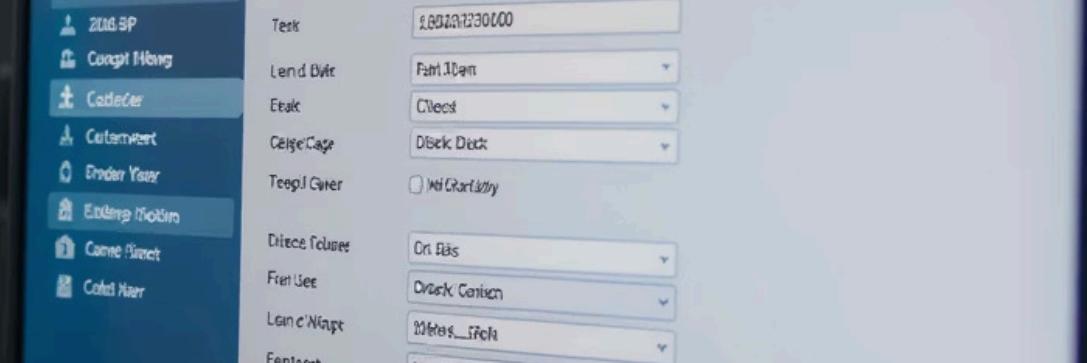


Esquema Visual



Lectura del diagrama, paso a paso

1. Seleccionas UEFI + GPT (recomendado) o Legacy + MBR por compatibilidad.
2. En UEFI, creas la ESP (FAT32), la raíz y, si procede, /home y swap.
3. Asignas filesystems adecuados al uso.
4. Formateas, instalas el SO y pasas a configuración inicial.
5. Tras el primer arranque, conectas a la red y aplicas actualizaciones.



Caso de Estudio – Proxmox VE: particionado para servidores robustos

Contexto

Proxmox VE es una plataforma de virtualización (KVM + LXC) usada en pymes y entornos profesionales. Su instalador ofrece elecciones de disco, sistema de archivos y opciones avanzadas que impactan directamente en rendimiento, resiliencia y administración.

Estrategia

- **Selección de FS:** Proxmox permite EXT4 (simplicidad y rendimiento sobrio) o ZFS (snapshots, compresión, verificación de integridad con checksums y posibilidad de espejado/RAID-Z).
- **Disposición del almacenamiento:** con ZFS sobre dos o más discos puedes configurar mirror o RAID-Z, alcanzando tolerancia a fallos sin controladoras RAID dedicadas. Para laboratorios con un solo SSD, EXT4 puede ser suficiente; para producción, ZFS aporta consistencia y recuperación ante bitrot.
- **Parámetros finos:** ajustas swap, tamaño del volumen raíz, y la política de memoria (ARC en ZFS) considerando la RAM para evitar que la caché del FS compita con las VMs.

Resultado

- Con ZFS + snapshots programados, recuperar el estado de una VM o un contenedor es cuestión de segundos, y la integridad de los datos mejora gracias a checksums y self-healing en espejos.
- En entornos con muchas VMs de bases de datos, elegir adecuadamente el aligranulado (recordsize en ZFS, bloc sizes) y la compresión puede reducir I/O y ahorrar espacio.
- En despliegues edge o con un solo disco, EXT4 reduce complejidad y consumo de RAM, facilitando administración y diagnósticos.

Conclusión: Proxmox demuestra que el particionado y el filesystem no son meros trámites; son decisiones arquitectónicas que determinan la fiabilidad, el rendimiento y la velocidad de recuperación del servicio.

Herramientas y Consejos

Planifica tu esquema con un objetivo claro

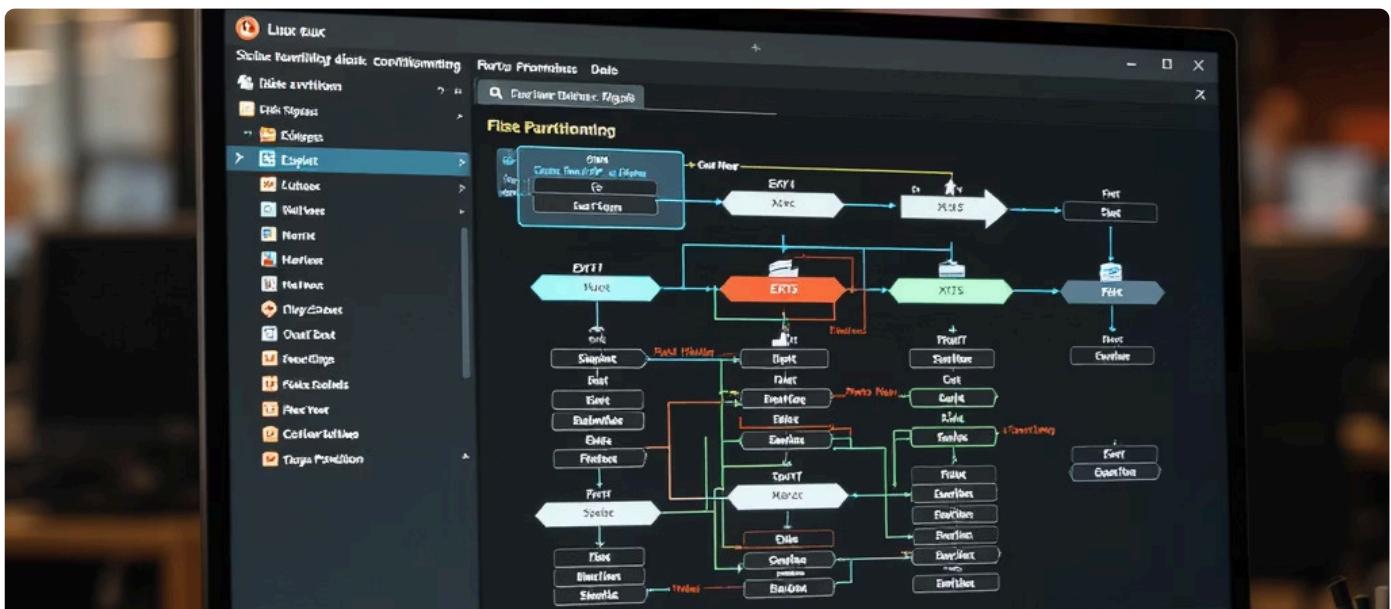
- **Escritorio Linux:** / (30–50 GB) + /home (resto) + swap (1–8 GB según RAM y hibernación).
- **Servidor:** separa /var si esperas logs intensivos o colas (e.g., correo, bases de datos), y considera LVM para ampliar volúmenes sin reinstalar.
- **Dual-boot:** mantén UEFI + GPT para ambos SO; crea ESP única (no duplique) y desactiva el Fast Startup de Windows para evitar bloqueos de NTFS.

Elige el sistema de archivos por caso de uso

- **EXT4:** seguro para casi todo en Linux, con buena compatibilidad y herramientas maduras.
- **XFS:** excelente en archivos grandes (backups, VMs), muy estable en servidores.
- **Btrfs/ZFS:** si valoras snapshots, compresión, checksum e instantáneas para rollback. Requieren más conocimiento y, en ZFS, más RAM.

GParted y compañía

- **GParted (live):** interfaz gráfica clara para crear, redimensionar y formatear particiones en Linux.
- **Windows:** Administración de discos y diskpart (CLI) para particionado; BitLocker si vas a cifrar.
- **Linux CLI:** lsblk, parted, gdisk (GPT), mkfs.ext4, mkfs.xfs, mkswap, btrfs, zpool/zfs para flujos automatizados.



Mitos y Realidades

 Mito: "Formatear un disco borra los datos para siempre."

FALSO. Un formateo rápido solo rehace metadatos/índices; los bloques pueden seguir conteniendo información recuperable. Para un borrado realmente seguro, usa sobrescritura completa, TRIM en SSD (y herramientas del fabricante) o métodos de cifrado + destrucción de claves.

 Mito: "Con mucha RAM no necesito swap."

FALSO. El swap sigue siendo útil para hibernación, estabilidad en picos de memoria y para que el kernel evite matar procesos críticos si aparece presión de RAM. Su tamaño y uso dependen del escenario, pero eliminarlo totalmente no siempre es buena idea.

Resumen Final

- Diseña el disco pensando en UEFI + GPT, con ESP y particiones separadas cuando conviene.
 - Elige filesystem según el caso de uso (EXT4/XFS para sencillez y rendimiento; Btrfs/ZFS para snapshots e integridad).
 - Define swap con criterio y contempla cifrado si manejas datos sensibles.
 - La configuración inicial (usuario, idioma, zona horaria y red) sella seguridad y usabilidad desde el primer día.

