

Técnico/a de Gestión de Sistemas y Tecnologías de la Información (A2)

SES (Servicio Extremeño de Salud)



opositaonline.com
“Elige tu destino · Cambia tu vida · Sé feliz”

TEMA 01

Administración básica de sistemas operativos GNU/Linux (Debian)

Contenidos:

- Instalación del sistema operativo
- Archivos y directorios
- Usuario root
- Configuración del arranque del sistema operativo
- Herramientas básicas de administración

TEMA 01.- ADMINISTRACIÓN BÁSICA DE SISTEMAS OPERATIVOS GNU/LINUX (DEBIAN)

CONTENIDOS

1. Introducción

1.1 Conceptos administrativos importantes

1.2 Tareas del Administrador

2. Instalación del sistema operativo

2.1 Esquema de particionamiento recomendado

3. Archivos y directorios

3.1 Archivos

3.2 Directorios

4. Usuario root

5. Configuración del arranque del sistema operativo

5.1 Niveles de ejecución

5.2 Proceso de arranque del sistema

6. Herramientas básicas de administración

ELIGE TU DESTINO • CAMBIA TU VIDA • SÉ FELIZ

6.1 Usuarios y grupos

6.2 Monitorización del sistema

6.3 Archivos de logs



OPOSITAONLINE

“Los obstáculos son esas cosas que las personas ven cuando dejan de mirar sus metas”
E. Joseph Cossman – Director de MK y emprendedor

1. Introducción

En todo **sistema Linux** existe un **USUARIO ADMINISTRADOR (ROOT)**, que **controla el funcionamiento completo del sistema**, tiene **acceso universal** y puede realizar cualquier operación con los datos y los dispositivos de la máquina.

Estos sistemas operativos **constan de estos 2 COMPONENTES: núcleo y Shell**:

- ↳ **Shell:** permite introducir comandos, ejecutar programas, etc. Además, el Shell es un “lenguaje de programación” que permite controlar como se están ejecutando los comandos.
- ↳ **NÚCLEO:** es la parte del sistema que interactúa con el hardware. Aporta servicios para la gestión de memoria, control de acceso a los periféricos, control y gestión del sistema de archivos, manejo de interrupciones, gestión de procesos de entrada/salida, etcétera

Es importante que el administrador de estos sistemas controle ambos componentes para las diferentes tareas que deberá llevar a cabo como gestor del sistema.

1.1. Conceptos administrativos importantes

Linux es un **sistema MULTIUSUARIO**, es decir, un **S.O. que permite que dos o más usuarios utilicen sus programas al mismo tiempo**.

Por otro lado, es un **MULTITAREA**, es decir, el **sistema permite que los usuarios estén ejecutando varias aplicaciones simultáneamente** utilizando la técnica de tiempo compartido.

Sistema de archivos jerárquico

Linux utilizan un **SISTEMA DE ARCHIVOS JERÁRQUICO** en forma de árbol invertido. La diferencia esencial frente a los sistemas tipo Windows es que **Linux no gestionan dispositivos (como un lector de DVD) de forma directa**. **Linux** gestiona los **DISPOSITIVOS** como si fueran **FICHEROS**, de tal forma que cuando estemos accediendo al fichero asociado a un lector de DVD, en realidad lo que estarás haciendo es acceder a la información contenida en el DVD.

En el **sistema de ficheros** de **Linux** (y similares), existen **varias SUBJERARQUÍAS** de **DIRECTORIOS** que poseen **múltiples y diferentes funciones** de almacenamiento y organización en todo el sistema. Estos directorios pueden clasificarse en:

- ↳ **ESTÁTICOS:** contiene **archivos que no cambian sin la intervención del administrador** (root), sin embargo, pueden ser leídos por cualquier otro usuario: `/bin, /sbin, /opt, /boot, /usr/bin...`
- ↳ **DINÁMICOS:** contiene **archivos que son cambiantes, y pueden leerse y escribirse** (algunos sólo por su respectivo usuario y el root). Contienen configuraciones, documentos, etc. Para estos directorios, es recomendable una copia de seguridad con frecuencia, o mejor aún, deberían ser montados en una partición aparte en el mismo disco, como, por ejemplo, montar el directorio `/home` en otra partición del mismo disco, independiente de la partición principal del sistema; de esta forma, puede repararse el sistema sin afectar o borrar los documentos de los usuarios: `/var/mail, /var/spool, /var/run, /var/lock, /home...`
- ↳ **COMPARTIDOS:** contiene **archivos que se pueden encontrar en un ordenador y utilizarse en otro, o incluso compartirse entre usuarios**.



- ☞ **RESTRINGIDOS:** contiene ficheros que no se pueden compartir, solo son modificables por el administrador: /etc, /boot, /var/run, /var/lock...

Herramientas gráficas y línea de comandos

Cualquiera de las distribuciones de GNU/Linux suele aportar **HERRAMIENTAS** del tipo **LÍNEA DE COMANDOS (textuales) O GRÁFICAS** que complementan las anteriores y simplifican en mayor o menor medida la **administración de las tareas**. Pero hay que tener en cuenta una serie de aspectos:

- a. Estas herramientas son una interfaz más o menos elaborada de las herramientas básicas de línea de comandos y los correspondientes ficheros de configuración.
- b. Normalmente no ofrecen todas las prestaciones o configuraciones que pueden realizarse a bajo nivel.
- c. El uso de estas herramientas oculta, a veces completamente, el funcionamiento interno del servicio o tarea.
- d. Estas herramientas son útiles en la mejora de la productividad.
- e. Puede darse el caso de que la tarea puede ser tan compleja, o necesitar tantos parámetros, o generar tantos datos, que se vuelve imposible controlarla de forma manual. En estos casos, las herramientas de alto nivel pueden ser muy útiles y volver practicables algunas tareas que de otra manera son difíciles de controlar. Por ejemplo, dentro de esta categoría entrarían las herramientas de visualización, monitorización, y resumen de actividades o servicios complejos.
- f. En la automatización de tareas, estas herramientas (de más alto nivel) pueden no ser las adecuadas.

Tenemos que saber valorar estas herramientas extra según la valía que tengan para nuestras tareas.

ELIGE TU DESTINO • CAMBIA TU VIDA • SÉ FELIZ

Shell scripts

Tal y como ya ha sido mencionado, el SHELL no sólo es un intérprete de comandos sino que puede utilizarse también como un lenguaje de programación, el lenguaje del shell. Un programa escrito con el lenguaje de programación de shell, se denomina GUION O SCRIPT DE SHELL, o programa de shell.

El LENGUAJE de SHELL es un lenguaje de programación de alto nivel que permite ejecutar secuencias de órdenes, seleccionar operaciones alternativas dependiendo del resultado de pruebas lógicas y repetir acciones del programa.

Es importante que un administrador maneje con soltura este lenguaje, pues le será de gran utilidad para la automatización de muchas tareas relacionadas con la gestión del sistema, de usuarios y grupo,...

Variables de entorno y variables de shell

El shell dispone de un mecanismo para DEFINIR VARIABLES que se pueden utilizar para almacenar información usada por los programas del sistema o para uso propio. Algunas variables las utiliza el propio shell u otros programas de Linux. Se pueden definir otras para ser usadas por el propio usuario. Muchas utilidades Linux, incluyendo el shell, necesitan información acerca del usuario y de qué pretende hacer para poder realizar bien el trabajo.

Por ejemplo, para ejecutar un comando o programa en Linux, es necesario especificar el nombre absoluto del programa o comando, o de lo contrario el sistema no sabrá qué comando ejecutar. En vez de forzar al usuario a proporcionar esa información cada vez que teclea un comando, Linux utiliza las variables de entorno como un mecanismo con el que usuario pueda almacenar esa información y despreocuparse de ella. En el caso que nos ocupa, la variable de entorno PATH, lista todos aquellos directorios que almacenan comandos o programas que el usuario desea ejecutar. Cuando se invoca a un comando, el shell recorre cada directorio de la variable PATH para buscar ese comando.

Probablemente, Linux no necesitaría una variable de entorno si todos los comandos estuvieran en el mismo directorio. La DIFERENCIA entre variables de entorno y variables de shell, es que las VARIABLES DE SHELL son locales a una shell, mientras que una VARIABLE DE ENTORNO se hereda por cualquier otro programa que se está ejecutando. Las variables de shell son útiles únicamente en el shell actual, pero las variables de entorno son exportadas automáticamente quedando disponibles de forma global. Por ejemplo, las variables de shell pueden estar disponibles únicamente a un script particular en el que están definidas, mientras que las variables de entorno se utilizan por cualquier script de shell, utilidad de correo, editor, etc. que lo invoque.

VARIABLES DE ENTORNO

Las variables de entorno se asignan del siguiente modo:

```
$ setenv NOMBRE valor          # En el shell C
$ NOMBRE = valor; export NOMBRE # En los shell de Bourne o Korn
```

ELIGE TU DESTINO • CAMBIA TU VIDA • SÉ FELIZ

Por convenio, las variables de entorno se denotan en mayúsculas. El usuario puede crear sus propias variables de entorno o puede utilizar variables de entorno predefinidas. Algunas de estas variables predefinidas son las siguientes:

- ↳ **SHELL**: Describe el shell que interpretará cualquier comando que introduzca. En la mayoría de los casos, será bash de forma predeterminada, pero se pueden establecer otros valores si prefiere otras opciones.
- ↳ **TERM**: Especifica el tipo de terminal a emular cuando se ejecuta el shell. Se pueden emular diferentes terminales de hardware para diferentes requisitos de funcionamiento. Sin embargo, generalmente no tendrá que preocuparse por esto.
- ↳ **USER**: El usuario que inició sesión actualmente.
- ↳ **PWD**: El directorio actual de trabajo.
- ↳ **OLDPWD**: El directorio anterior de trabajo. Esto se mantiene en el shell para volver a su directorio anterior ejecutando cd -.
- ↳ **LS_COLORS**: Define los códigos de color que se utilizan para añadir de forma opcional un resultado de color al comando ls. Esto se utiliza para distinguir diferentes tipos de archivos y proporcionar más información al usuario de forma rápida.
- ↳ **MAIL**: La ruta al buzón del usuario actual.
- ↳ **PATH**: Una lista de directorios que el sistema comprobará cuando busque comandos. Cuando un usuario escriba un comando, el sistema comprobará los directorios en este orden para el ejecutable.
- ↳ **LANG**: Las configuraciones actuales de idioma y localización, incluida la codificación de caracteres.
- ↳ **HOME**: El directorio principal del usuario actual.
- ↳ **_**: El comando más reciente ejecutado anteriormente.



Para ver el contenido de estas variables de entorno solo tenemos que usar el comando `env` o `printenv` o también mediante la orden `echo $nombre variable`.

VARIABLES DE SHELL

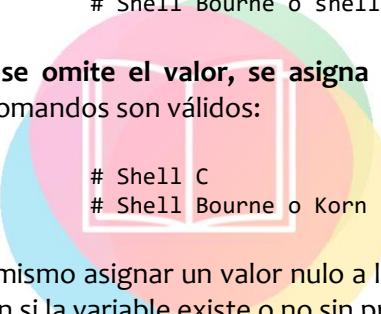
Las variables de shell vienen a ser una GENERALIZACIÓN DE LAS VARIABLES DE ENTORNO. Las variables de shell pertenecen al shell y se comportan de forma similar a como lo hacen las variables en un lenguaje de programación como C o Fortran. En términos de programación puede pensarse que las variables de entorno son variables globales mientras que las variables de shell son variables locales. Por convención las variables de shell se denotan en minúsculas. Para asignar valores a variables de shell puede utilizarse el comando:

```
$ set nombre=valor      # Shell C
$ nombre=valor          # Shell Bourne o shell Korn
```

Como caso especial, si se omite el valor, se asigna un valor null a la variable. Por ejemplo, los siguientes comandos son válidos:

```
$ set nombre
$ nombre=          # Shell C
                  # Shell Bourne o Korn
```

Obsérvese que no es lo mismo asignar un valor nulo a la variable que borrar la variable. Algunos programas miran si la variable existe o no sin preocuparse de su contenido. Si se desea que la shell se olvide de que una variable existió, se utiliza el comando `unset`.



OPOSITAONLINE
 ELIGE TU DESTINO • CAMBIA TU VIDA • SÉ FELIZ

1.2. Tareas del Administrador

El administrador de cualquier tipo de servidor debe ser una persona especializada, que conozca lo mejor posible sus equipos, sus aplicaciones y sus usuarios; manteniéndose al día en los avances tecnológicos, en las revisiones y parches de los programas instalados y en las necesidades de su empresa.

Planificación y previsión de necesidades

Una de las funciones principales en la administración de sistemas informáticos es la planificación detallada de las TAREAS DE GESTIÓN, lo que puede evitar sorpresas desagradables en el momento de ejecutarlas.

El analista de sistemas tiene la obligación de asesorar al personal administrativo de su empresa sobre las necesidades tecnológicas en la adquisición de material informático, estimando los recursos que precisen los usuarios, en relación con las posibilidades económicas de la empresa.

Una vez recibido el equipo debe realizarse un plan de instalación, en el que se incluya, al menos la siguiente información:

- ↳ Documentación y estudio de los recursos disponibles.
- ↳ Previsión de posibles ampliaciones.

- ↳ **Relleno de solicitud de alta en la red informática corporativa** y activación de los parámetros de conexión.
- ↳ **Documentación de necesidades del entorno de operación** (SAI, aire acondicionado, etc.).
- ↳ **Documentación sobre registro, configuración, instalación y actualización del sistema operativo**, de las aplicaciones requeridas y de los programas propios, de acuerdo con los servicios que debe prestar el nuevo equipo.
- ↳ **Creación y publicación de solicitudes de apertura y modificación de cuentas de usuarios**, de **instalación de programas**, de mejora de recursos, etc.

Documentación

El responsable del sistema se compromete a realizar también **documentación interna** que debe describir las siguientes necesidades:

- ↳ **Registro actualizado de los usuarios y grupos del sistema**.
- ↳ **Políticas de utilización y permisos** para cada grupo de usuarios.
- ↳ **Descripción de los procedimientos comunes que deben ejecutar los operadores** del **sistema** (copias de seguridad, gestión de cuentas, informes, etc.).
- ↳ **Registro completo y actualizado de los cambios** en la configuración del servidor (sistema operativo, aplicaciones, ficheros, etc.).
- ↳ **Recogida periódica y archivado de datos** sobre el **rendimiento del sistema** y de sus componentes.

Automatización

El personal informático de una empresa ha de ejecutar periódicamente las funciones definidas en el plan de actuación. El programador necesita **automatizar la mayoría de estos procedimientos repetitivos** para evitar errores tipográficos o conceptuales, y para mejorar el tratamiento general de las aplicaciones.

En cada servidor **deben automatizarse, al menos, las siguientes tareas**:

- ↳ **Comprobación del espacio libre** en los **discos**.
- ↳ **Gestión de cuentas de usuarios y revisión periódica de las cuotas de disco**.
- ↳ **Procedimientos** para crear, comprobar y restaurar **copias de seguridad**, según el plan de actuación.
- ↳ **Comprobación y registro del rendimiento** general del **sistema y de la red** informática.
- ↳ **Trabajos específicos** (informes, gestión de servicios, creación de documentación, etc.).
- ↳ **Creación de alertas de seguridad** (comprobación de cambios, detección de intrusos, etc.).

Para poder automatizar estas tareas Linux pone a nuestra disposición comandos como at, cron, crontab...

at

at es una herramienta que **permite PROGRAMAR la EJECUCIÓN de uno o varios PROGRAMAS en un momento FUTURO**.



La sintaxis normal de at es

```
at [hora:minuto [dia.mes.año]]
```

Las instrucciones a realizar son leídas por la entrada estándar (stdin) y se ejecutan en el momento indicado. Como es habitual, la lista de instrucciones se debe finalizar con el carácter EOF, usualmente Control+D. Si solo se indica la hora/minuto, se ejecutará cuando se alcance ésta (en el día actual o el siguiente). Si se especifica una fecha, necesariamente se especificará una hora.

Los resultados que muestre el programa por la salida estándar (stdout) o la salida de error (stderr) se enviarán por correo electrónico al usuario que invocó at.

- ↳ La instrucción at -l permite listar las ejecuciones programadas de at.
- ↳ La instrucción at -d permite borrar una o varios trabajos (igual que -r).
- ↳ La instrucción at -f fichero hora:minuto ejecutará en el momento programado las instrucciones contenidas en el fichero indicado, en vez de leerlas por la entrada estándar.

cron y crontab

El nombre cron viene del griego chronos que significa “tiempo”. En el sistema operativo Linux, cron es un ADMINISTRADOR regular de PROCESOS EN SEGUNDO PLANO (demonio) que ejecuta procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los PROCESOS que deben ejecutarse Y LA HORA en la que deben hacerlo SE ESPECIFICAN en el fichero crontab.

El demonio cron se inicia desde /etc/rc.d/ o /etc/init.d dependiendo de la distribución. cron se ejecuta en el background, revisa cada minuto la tabla de tareas crontab /etc/crontab o en /var/spool/cron en búsqueda de tareas que se deban cumplir. Como usuario podemos agregar comandos o scripts con tareas a cron para automatizar algunos procesos. Esto es útil por ejemplo para automatizar la actualización de un sistema o un buen sistema de copias de seguridad.

crontab es un simple ARCHIVO DE TEXTO que guarda una LISTA DE COMANDOS A EJECUTAR en un momento especificado por el usuario. crontab verificará la fecha y hora en que se debe ejecutar el script o el comando, los permisos de ejecución y lo realizará en el background. Cada usuario puede tener su propio archivo crontab, de hecho el /etc/crontab se asume que es el archivo crontab del usuario root, cuando los usuarios normales (e incluso root) desean generar su propio archivo de crontab, entonces utilizaremos el comando crontab.

crontab es la manera más sencilla de administrar tareas de cron en sistemas multiusuario, ya sea como simple usuario de sistema o usuario root.

Agregar tareas a crontab

Ejecutamos la edición del crontab con crontab -e. El archivo crontab lucirá algo así.

```
# m h dom mon dow user command
```

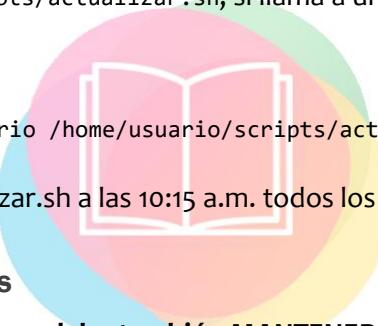
donde:

- ↳ **m** corresponde al **minuto** en que se va a ejecutar el script, el valor va de 0 a 59
- ↳ **h** la **hora exacta**, se maneja el **formato de 24 horas**, los valores van de 0 a 23, siendo 0 las 12:00 de la medianoche.
- ↳ **dom** hace referencia al **día del mes**, por ejemplo, se puede especificar 15 si se quiere ejecutar cada día 15
- ↳ **mon** indica el mes
- ↳ **dow** significa el **día de la semana**, puede ser numérico (0 a 7, donde 0 y 7 son domingo) o las **3 primeras letras del día en inglés**: mon, tue, wed, thu, fri, sat, sun.
- ↳ **user** define el **usuario** que va a ejecutar el comando, puede ser root, u otro usuario diferente siempre y cuando tenga permisos de ejecución del script.
- ↳ **command** se refiere al **comando** o a la ruta absoluta del script a ejecutar, ejemplo: /home/usuario/scripts/actualizar.sh, si llama a un script este debe ser ejecutable

Por ejemplo:

```
15 10 * * * usuario /home/usuario/scripts/actualizar.sh
```

Ejecutará el script actualizar.sh a las 10:15 a.m. todos los días



Informar a los usuarios

El **administrador de sistema** debe también **MANTENER INFORMADOS A SUS USUARIOS** y darles unas guías de operación y buen uso, lo que puede evitar errores provocados por desconocimiento.

TAMBIÉN es necesario **INFORMAR SOBRE LOS CAMBIOS** que pueden afectar **a cada grupo** de usuarios, indicando la siguiente información:

- ↳ **La naturaleza de los cambios** que van a realizarse en el sistema y su evolución temporal.
- ↳ **Cuándo** se realizará cada modificación.
- ↳ **Qué resultados se esperan obtener** con la operación y cuáles son los obtenidos.
- ↳ **Tiempo estimado y tiempo real de la duración de la operación**.
- ↳ **Impacto posible sobre los usuarios** (nueva configuración, parada del sistema, etc.).
- ↳ **Información de contacto** para recoger dudas y consultas.

Por otro lado, el encargado del sistema tiene la obligación de conocer profundamente el comportamiento general de sus usuarios, registrando sus consultas, sus sugerencias y los datos de rendimiento y utilización de recursos. Esto permite ofrecer una mejor calidad en los servicios ofertados.

Control de la seguridad del sistema

Dependiendo del tipo de información tratada por el sistema, el **administrador debe DEFINIR sus POLÍTICAS DE SEGURIDAD, tanto para el servidor, como para la red corporativa**, ya que los usuarios tienen derecho a la privacidad e integridad de sus datos.

Deben ponerse los medios para evitar posibles ataques o fallos informáticos que afecten –o incluso paralicen– el funcionamiento normal de la máquina.

Nunca hay que tener la presunción de que un sistema es completamente seguro o de que sólo puede ser atacado desde fuera. Por ello, el **SUPERUSUARIO** debe realizar las siguientes **OPERACIONES**:



- ↳ **Activar y revisar los registros históricos de incidencias.**
- ↳ **Realizar revisiones periódicas sobre posibles cambios no deseados en el sistema.**
- ↳ **Instalar aplicaciones y dispositivos que protejan a los servidores y a la red** informática (sistemas de detección de intrusos, cortafuegos, filtros, lectores de tarjetas de acceso, etc.).

2. Instalación del sistema operativo

Aunque la instalación de cualquier distribución Linux es muy similar, pondremos como **EJEMPLO** la **INSTALACIÓN DE DEBIAN**.

Debian es una de las mejores distribuciones Linux que podemos encontrar, y una muestra de ello es que la **mayor parte de las distros** (entre ellas, Ubuntu) **se basan en ella**. Esta distribución cuenta con un soporte sobresaliente, una gran comunidad y es una **de las más estables** que podemos encontrar, tanto en PC como en servidor. Sin embargo, es cierto que es algo más complicada de poner en marcha, instalar y configurar al no ser tan simple e intuitiva como, por ejemplo, la distro de Canonical.

Por ello, vamos a ver paso a paso cómo instalar y poner en marcha este Linux.

1. Descargar y preparar la instalación

Para poder instalar este Linux, lo primero que tenemos que hacer es **descargar la última versión**. La descarga, por supuesto, es totalmente gratuita.

A la hora de bajar Debian vamos a poder elegir **ENTRE VARIAS IMÁGENES**.

- ↳ Por un lado vamos a tener la «**NETINST**». Esta es una **ISO muy pequeña pensada para descargarse de forma rápida y no ocupar mucho espacio en el PC** si queremos guardarla. Viene con **lo justo para arrancar y comenzar la instalación**, pero necesitaremos una conexión a Internet para poder descargar todos los paquetes y poder completar la instalación.
- ↳ Por otro lado tendremos la «**FULL**». Esta ISO **trae todos los paquetes necesarios para poder instalar y poner en marcha Debian sin necesidad de Internet**. Recomendada para ordenadores con conexión lenta o sin conexión.
- ↳ Y, por último, tenemos la edición «**LIVE**». Esta está **pensada** para aquellos **usuarios que quieran probar Debian antes de instalarla en su ordenador**. Desde el propio sistema cargado en la RAM vamos a poder instalarlo a través del instalador Calamares incluido.

Una vez **descargada la imagen** que queramos, lo que debemos hacer es **GRABARLA** a un **CD, DVD o, mejor, a una memoria USB**.

2. Instalar Debian

Nosotros vamos a usar la versión «netinst» de esta distro, aunque el proceso con la «full» es exactamente el mismo. **Arrancamos nuestro ordenador con dicha ISO y podremos ver el menú de arranque de nuestra distro.**

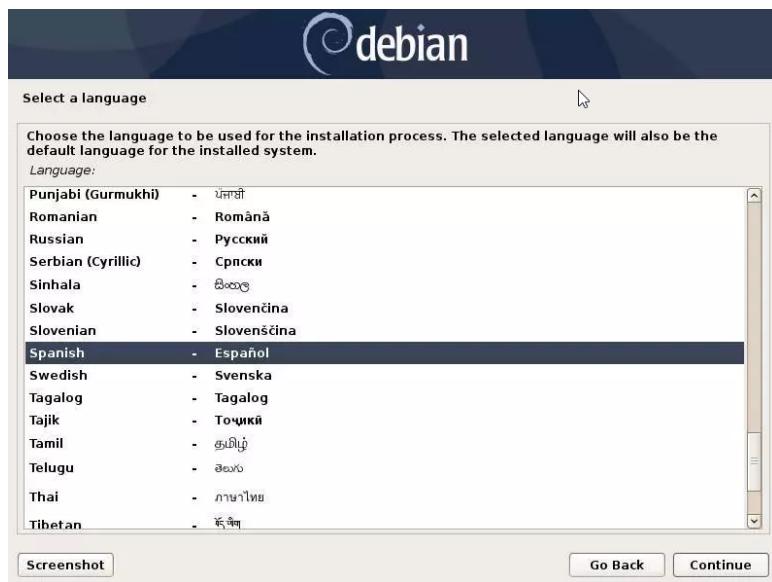


Aquí nos vamos a encontrar las DISTINTAS OPCIONES DE ARRANQUE QUE TENEMOS. Debian nos ofrece básicamente dos tipos de instalación diferentes. Por un lado tendremos la instalación CLÁSICA, en MODO TEXTO, Y por otro una instalación GRÁFICA guiada CON UN ASISTENTE. Nosotros vamos a elegir esta segunda, ya que es más sencilla. Seleccionamos «Graphical Install», y esperamos a que cargue el instalador.

3. Empezando la instalación

ELIGE TU DESTINO • CAMBIA TU VIDA • SÉ FELIZ

Lo primero que tendremos que hacer será SELECCIONAR EL IDIOMA de nuestra distro, el cual se aplicará también al asistente de instalación.

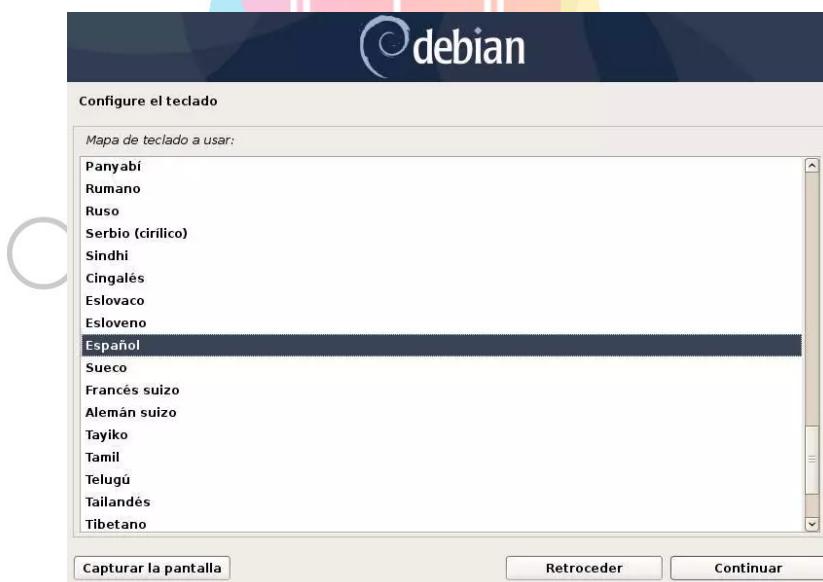


El siguiente paso será ELEGIR NUESTRA UBICACIÓN. Gracias a ella Debian podrá configurar automáticamente la zona horaria y nuestra localización.





Y también elegiremos el IDIOMA de nuestro TECLADO.



Ahora que el idioma y el teclado de nuestra distro están configurados, podemos continuar.

4. Detectar y configurar el hardware

Después de las primeras configuraciones, Debian dedicará un tiempo a ¹volcar los datos de nuestro medio de instalación (del USB, del CD o de la ISO) al ordenador para poder acceder a ellos más rápidamente. También aprovechará para ²detectar todo el hardware de nuestro ordenador, configurarlo para que funcione y, por supuesto, ³conectarnos a Internet.



Este proceso puede tardar hasta unos minutos, dependiendo de la potencia del PC.

También podremos DARLE UN NOMBRE A NUESTRO PC (para identificarlo mejor en red) y, además, UNIRNOS A UN DOMINIO, si trabajamos con esta característica.



5. Crear y proteger usuarios al instalar Debian

Llegados a este punto, lo que tenemos que hacer es **CREAR** los **USUARIOS Y PROTEGER** adecuadamente las **CUENTAS**. Lo primero que nos pedirá Debian es introducir la **CONTRASEÑA** que queremos darle a la cuenta de **SUPERUSUARIO, O ROOT**. Esta contraseña debe ser larga y robusta, ya que cualquiera que tenga acceso a ella tendrá control total sobre el equipo.



Después de haber creado la cuenta de superusuario es hora de **CREAR EL PRIMER USUARIO**. Debian nos pedirá que introduzcamos en primer lugar el nombre completo del mismo.

ELIGE TU DESTINO • CAMBIA TU VIDA • SÉ FELIZ



Y el nombre de usuario que le queremos dar. Y en tercer lugar, la contraseña de dicho usuario. Esta contraseña también debe ser segura si queremos evitar que personas no autorizadas se conecten al PC sin permiso.



6. Otras configuraciones del sistema

A continuación, lo que haremos será elegir la **UBICACIÓN DE NUESTRA ZONA HORARIA**, ya que así se podrá ajustar el reloj automáticamente en función de ella.

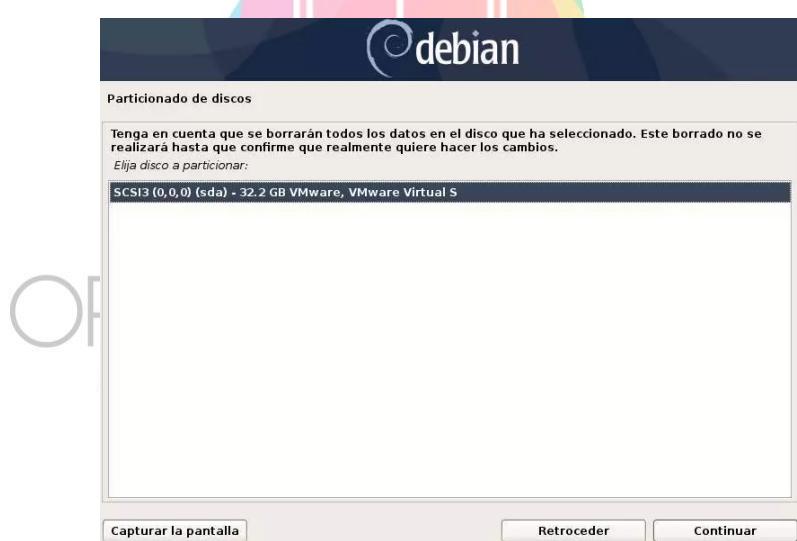


Ahora le ha llegado el turno al disco duro. **Debian**, como cualquier otro Linux, **utiliza una serie de PARTICIONES Y PUNTOS DE MONTAJE necesarios para poder guardar todos los datos**. Si vamos a dedicar todo el disco podemos usar el método guiado para que el proceso de creación de las particiones sea lo más sencillo posible. Si somos usuarios avanzados, y queremos tener más control sobre las particiones, entonces podemos usar el método manual. También podemos elegir alguna de las otras dos opciones si lo que buscamos es aplicar un cifrado al disco para protegerlo.

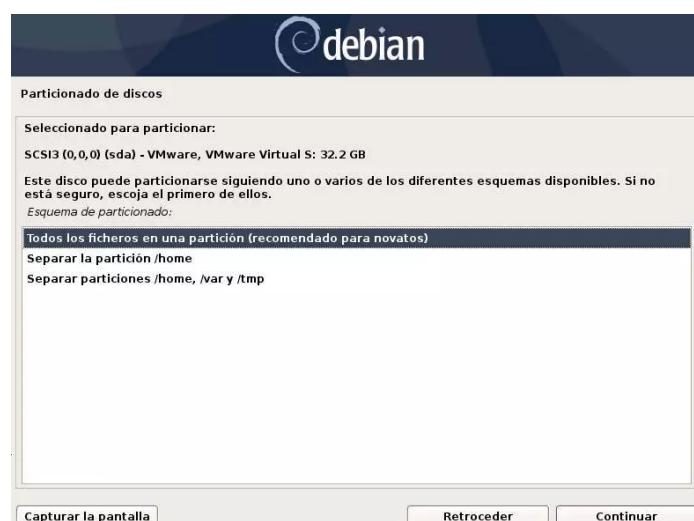




Nosotros vamos a usar el método guiado para usar todo el disco. Lo primero será **SELECCIONAR el DISCO DURO O SSD donde vamos a INSTALAR nuestra distribución.**



En el siguiente paso elegiremos la **ESTRUCTURA DE PARTICIONES QUE VAMOS A USAR.** Podemos instalar todo el sistema en una partición, separar la /home en una partición diferente, o crear particiones independientes para /var y /tmp.



El asistente de instalación de Debian hará los cálculos correspondientes, y podremos ver un **RESUMEN CON TODOS LOS CAMBIOS** que se van a realizar.



Si está todo correcto, **aceptamos los cambios y, una vez particionado el disco, COMENZARÁ LA INSTALACIÓN COMO TAL.**



7. Comenzando la copia de archivos para instalar Debian

Debemos esperar a que el sistema termine de instalarse en el PC. Durante el proceso se descargará los paquetes que faltan al usar la imagen netint, y se configurará la distro para poder funcionar.

Al acabar la copia de archivos, el asistente nos preguntará si tenemos otro CD o DVD de software que queramos instalar al mismo tiempo. Esto era bastante frecuente en el



pasado, pero ahora ya está en desuso. Por tanto, como no tendremos otro medio de instalación, elegiremos la opción «No».

Ahora Debian nos pedirá que elijamos la **UBICACIÓN DEL SERVIDOR DE LOS REPOSITORIOS desde donde se descargarán los paquetes y las actualizaciones**. Es recomendable que este servidor esté lo más cerca posible de nuestro país para que la velocidad sea lo mayor posible.



Y también tendremos que **ELEGIR EL SERVIDOR DESDE DONDE BAJAR TODO**. Generalmente, el que viene por defecto (deb.debian.org) suele ser una buena elección.



Por último, **si usamos un proxy o alguna configuración de red extraña, tendremos que introducirla.**

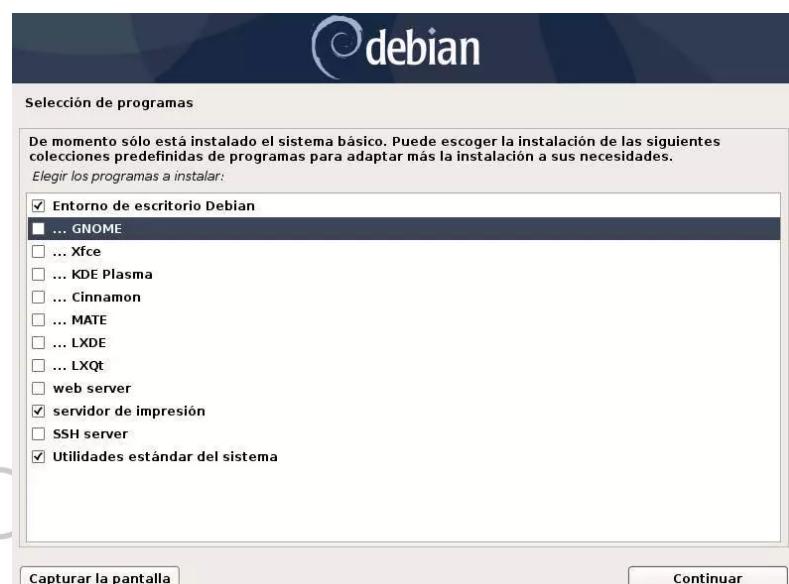
Con todo esto, el asistente de instalación de Debian se conectará a los repositorios y actualizará la base de datos de software, descargando otras dependencias o listas que no se encuentren en el sistema.

Y, no menos importante, podemos configurar si queremos compartir datos anónimos de uso. Por defecto, no.

Ya tenemos nuestra distribución instalada y configurada. Pero antes de terminar, podremos elegir el software que queremos instalar.

8. Instalando software adicional

Ahora que ya tenemos la distro instalada en el PC, antes de empezar a trabajar con ella **podemos ELEGIR QUÉ PROGRAMAS QUE QUEREMOS INSTALAR**. Debian nos va a permitir elegir el entorno de escritorio que queremos instalar, si queremos instalar algún servidor concreto o las utilidades estándar del sistema.



Hacemos clic sobre continuar, y **comenzará la descarga e instalación de los paquetes que hayamos seleccionado**. Este proceso puede tardar bastante tiempo, por lo que debemos esperar a que acabe.

Cuando acabe el proceso, ya estaremos en el **PASO FINAL**. Lo único que nos quedará por hacer es elegir si queremos **INSTALAR EL GESTOR DE ARRANQUE GRUB**, e indicar **EN QUÉ UNIDAD** queremos instalarlo.



Ya hemos terminado de instalar Debian. Ya solo nos queda dejar que finalice el asistente, reiniciar el ordenador y podremos empezar a disfrutar de nuestra nueva distro Linux.



2.1. Esquema de particionamiento recomendando

SE RECOMIENDA crear **SISTEMAS DE ARCHIVOS SEPARADOS** en los siguientes puntos de montaje. Sin embargo, si es necesario, también puede crear los sistemas de archivos en los puntos de montaje /usr, /var, y /tmp.

- ↳ /boot
- ↳ / (raíz)
- ↳ /home
- ↳ swap
- ↳ /boot/efi

/boot - tamaño recomendado al menos 1 GiB

La **partición** montada en /boot **CONTIENE EL KERNEL del sistema operativo**, que permite a su sistema arrancar Linux, **JUNTO CON los archivos utilizados durante el proceso de arranque**. Debido a las limitaciones de la mayoría de los firmwares, se recomienda crear una pequeña partición para contenerlos. En la mayoría de los escenarios, una partición de arranque de 1 GiB es adecuada. A diferencia de otros puntos de montaje, **el uso de un volumen LVM para /boot no es posible** - /boot debe estar ubicado en una partición de disco separada.

NOTA: normalmente, la partición /boot es creada automáticamente por el programa de instalación. Sin embargo, si la partición / (raíz) es mayor de 2 TiB y se utiliza (U)EFI para el arranque, es necesario crear una partición /boot separada que sea menor de 2 TiB para arrancar la máquina con éxito.

NOTA: **Si tiene** una tarjeta **RAID**, tenga en cuenta que **algunos tipos de BIOS no admiten el arranque desde la tarjeta RAID**. En tal caso, la **partición /boot debe crearse en una partición fuera de la matriz RAID**, como por ejemplo en un disco duro separado.

root - tamaño recomendado de 10 GiB

Aquí es donde **se encuentra "/"**, o el **DIRECTORIO RAÍZ**. El directorio raíz es el **nivel superior de la estructura de directorios**. Por defecto, **n** en la ruta en la que se escribe, por ejemplo, /boot o /home.

Aunque un sistema de archivos raíz de 5 GiB le permite realizar una instalación mínima, se recomienda asignar al menos 10 GiB para poder instalar todos los grupos de paquetes que desee.

Importante: **No confunda el directorio / con el directorio /root**. El directorio /root es el directorio principal del usuario raíz. El directorio /root se denomina a veces slash root para distinguirlo del directorio raíz.

/home - tamaño recomendado al menos 1 GiB

Para **almacenar los DATOS DEL USUARIO por separado de los datos del sistema**, cree un sistema de archivos dedicado para el directorio /home. Base el tamaño del sistema de archivos en la cantidad de datos que se almacenan localmente, el número de usuarios, etc. Puede actualizar o reinstalar Linux sin borrar los archivos de datos de usuario.