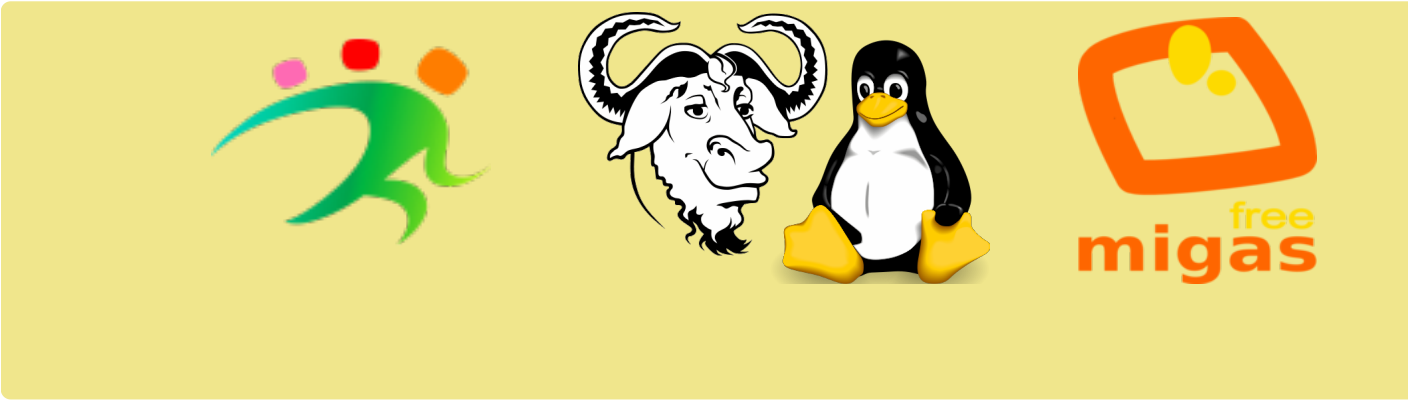
Software Libre

- Introducción al Curso
- Parte 1. Introducción al Software Libre
  - Presentación Contenidos Parte 1
  - ¿Conocimiento Libre?
  - ¿Qué es Software Libre?
- Parte 2. Distribuciones Libres: Vitalinux
  - Presentación Contenidos Parte 2
  - ¿Qué es Vitalinux?
  - Wiki de Vitalinux
  - Web de Soporte Vitalinux
  - Donde descargar Vitalinux
  - Como probar Vitalinux
- Parte 3. Aplicaciones Libres
  - Introducción a las Aplicaciones Libres
  - De dónde viene el Software
  - Cómo instalar software y ver sus fuentes
  - Aplicaciones comunes. Breve Guía
  - Créditos
- Créditos

Copyright 2026

Introducción al Curso



En el año 2014 la **Diputación General de Aragón *(DGA)*** inicia en modo pilotaje el **programa de Software Libre** con la finalidad de alcanzar los siguientes objetivos:

1. *Poder Reutilizar equipos antiguos o de bajas prestaciones*
2. *Alternativa a Windows 7/8* (MS deja de dar soporte a WXP y a Windows 7)
3. *Evitar el "pirateo" de Software privativo*
4. *Reducir los costes de licencias de Software*
5. *Fomentar el uso del Software Libre en Educación*
6. *Poder Inventariar todo el Software y Hardware*
7. *Facilitar labores de Gestión de Software gracias a Migasfree*

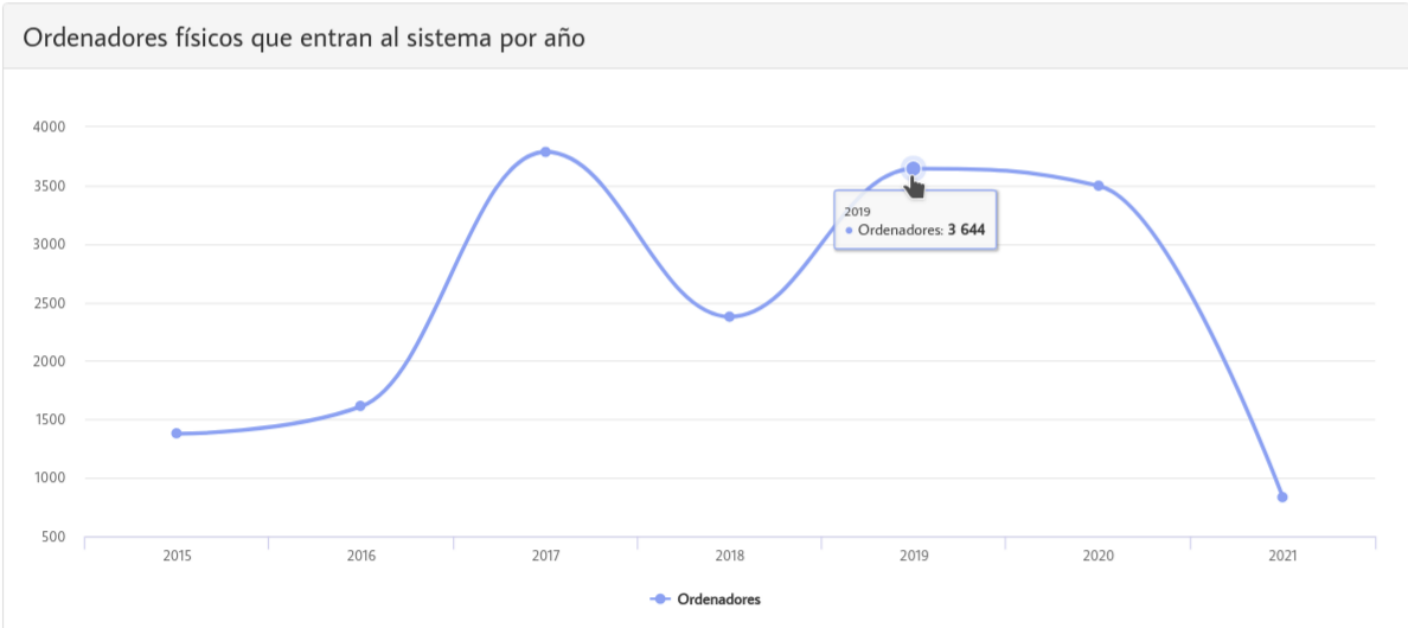
Como resultado de esa iniciativa surge el sistema operativo {{book.vitalinux}}, basado por completo en **código abierto**, con el cual podremos familiarizarnos a través del siguiente curso (URL del eBook)

Actualmente {{book.vitalinux}} se encuentra en su versión 2.0 con **5779** instalaciones realizadas, y con **11898** instalaciones de su versión anterior 1.0 (***dato extraidos de migasfree.educa.aragon.es el 22 de febrero del 2021***), y oficialmente es usado en más de 100 centros públicos de la Comunidad Autónoma de Aragón (***centros de Educación Infantil y Primaria, centros de Enseñanza Secundaria Obligatoria, Colegios Rurales Agrupados y Centros Públicos de Educación de Personas Adultas***): Mapa Oficial Centros Vitalinux

Antes de empezar el curso, y a modo de curiosidad mostraremos unas gráficas que nos informan del estado y uso de {{book.vitalinux}} y el Software Libre en nuestra Comunidad Autónoma.

Copyright 2026

Comenzaremos mostrando una gráfica que nos muestra cuantos equipos pasan a Vitalinux cada año. Como puede observarse, el mejor año hasta ahora fue el 2017 donde entraron a formar parte de la comunidad Vitalinux **3786 nuevos ordenadores**, sin desdeñar otros años como el 2019 donde lo hicieron **3644 nuevos ordenadores**. Señalar que en este año 2021, con tan sólo 2 meses de recorrido ya han migrado a Vitalinux **920 nuevos ordenadores** por lo que auguramos que será un buen año.



Otras gráficas interesantes pueden ser las que nos informe del número de equipos Vitalinux que se arrancan diariamente y por mes. Eso nos dará una idea del uso que se hace de Vitalinux en los centros educativos. Como podrá observarse los equipos Vitalinux en los centros no se arrancan todos los días (***salas de informática, equipos miniportátiles, equipos de aula, etc.***), además de poderse apreciar perfectamente la disminución increible que hubo desde abril del 2020 por la pandemia del coronavirus.

Copyright 2026





¡Migasfree es la clave!': Es muy importante señalar/recalcar que el éxito del ***Programa de Software Libre Vitalinux*** *es* ***Migasfree****, ya que sin este servicio de* ***gestión masiva de equipos de manera remota y desatendida, Vitalinux*** *no sería más que otro sistema operativo más. Más adelante veremos someramente todas las bondades de* ***Migasfree****.*

Parte 1. Introducción al Software Libre

Copyright 2026

Parte 1. Introducción al Software Libre

Presentación Contenidos Parte 1

¡¡Bienvenidos al Software Libre!!

En esta primera parte del curso de introducción al **Software Libre** se tratará de explicar mediante ejemplos reales las bondades de que un área de conocimiento sea libre y se detallará el concepto de libertad en el mundo del software.

Es recomendable ver el siguiente vídeo previamente para afianzar mejor los conceptos que se verán posteriormente.

¡¡Muchos ánimos y esperamos que el curso no te defraude!!

https://www.youtube.com/embed/1ZKo0oUgR8s

Copyright 2026

Parte 1. Introducción al Software Libre

¿Conocimiento Libre?

Podríamos copiar aquí la definición de **Conocimiento Libre** de la wikipedia, pero consideramos que no es necesario para llegar a comprender su importancia y bondades. Creemos que es mejor presentar ejemplos prácticos para su mejor comprensión. Además, con todo ello el **Conocimiento Libre** busca el beneficio colectivo y un mundo de oportunidades igualitario para todos.

La mejor forma de comprender qué es el conocimiento libre y sus beneficios es compararlo con el conocimiento cerrado o privativo. Para ello, a modo de ejemplo, se muestran dos productos aparentemente iguales pero que difieren en la información que aportan: **abierta** y **cerrada**.



image 2.2.1 - Comparación entre conocimiento abierto y cerrado

¿Cúal de los dos productos preferirías comprar? ¿Elegirías un producto sin saber los ingredientes que lo componen y procedimiento que se ha seguido para su elaboración?
En el mundo de software ocurre algo similar, hay productos software cuyos desarrolladores/creadores proporcionan una completa información de como ha sido construido el software y otros que no. Paradójicamente, el software cerrado está mucho más extendido que el software abierto en los equipos informáticos de sobremesa y portátiles, propiciado por una potente

maraña de intereses económicos, financieros y monetarios por parte de unas pocas compañías informáticas con hegemonía mundial que están interesadas en que ese panorama siga durante mucho tiempo así. El conocimiento libre permite que cualquiera tenga el conocimiento suficiente para poder reproducir y distribuir cualquier tipo de producto que sea libre, garantizando que los beneficios sean más contenidos y repartidos, impidiendo que sean sólo unas pocas las compañías que tengan su hegemonía. Esta claro que eso no interesa a aquellas empresas que ansien acaparar muchísimos beneficios.

Siguiendo con el símil anterior, y tratando de comprender el término **Conocimiento Libre**, ¿Qué sería una **Cocina libre o Cocina Abierta**? Una **cocina abierta** sería aquella donde tanto los ingredientes como el proceso de creación son de acceso público para que todo aquel que quiera **reproducirlo con exactitud** pueda hacerlo, dando libertad a poder modificar la lista de ingredientes o pasos a seguir en el proceso de creación, en el caso de considerar que puede mejorarse, y distribuir la nueva receta para que otros puedan disfrutar del nuevo plato confeccionado. Por contra, una cocina cerrada sería aquella donde ni la lista de ingredientes, ni el proceso a seguir puede hacerse pública, y por tanto, no esta permitida su distribución (***sería necesario adquirir una licencia para ello***) imposibilitando que públicamente pueda disfrutarse de ello, a menos que se pague.

Ejemplo de Conocimiento Libre: Cocina Libre

Supongamos que un cocinero Zaragozano crea un riquísimo y delicioso plato, con un éxito enorme entre la población y con una gran demanda de reservas en su restaurante para poder probarlo, decide ceder en formato libre la información de sus ingredientes y proceso de creación. ¿Qué beneficios colectivos aportaría?

- Permite que cualquier persona pudiera reproducir el plato en su casa y degustarlo sin necesidad de ir a ese restaurante. Este beneficiará sobre todo a aquellas personas que vivan en localidades no cercanas a la del restaurante.
- Permite que otros cocineros emprendedores de otras ciudades, a nivel nacional o mundial, puedan hacer uso de esa receta y tener el mismo éxito en sus restaurantes que está teniendo él.
- Permite que cualquier persona pueda modificar la receta y adaptar su elaboración a las necesidades de otros grupos de usuarios (***celiacos, vegetarianos, veganos, etc.***), ampliando de esta forma el grupo de usuarios que se benefician de ello.
- Permite por tanto mejorar y distribuir las mejoras en la receta para que la comunidad global se beneficie de ello: clientes para su degustación y restaurantes para obtener beneficios de ello.

Copyright 2026

- En definitiva, evita monopolizar los beneficios y garantiza el reparto y distribución de sus beneficios.

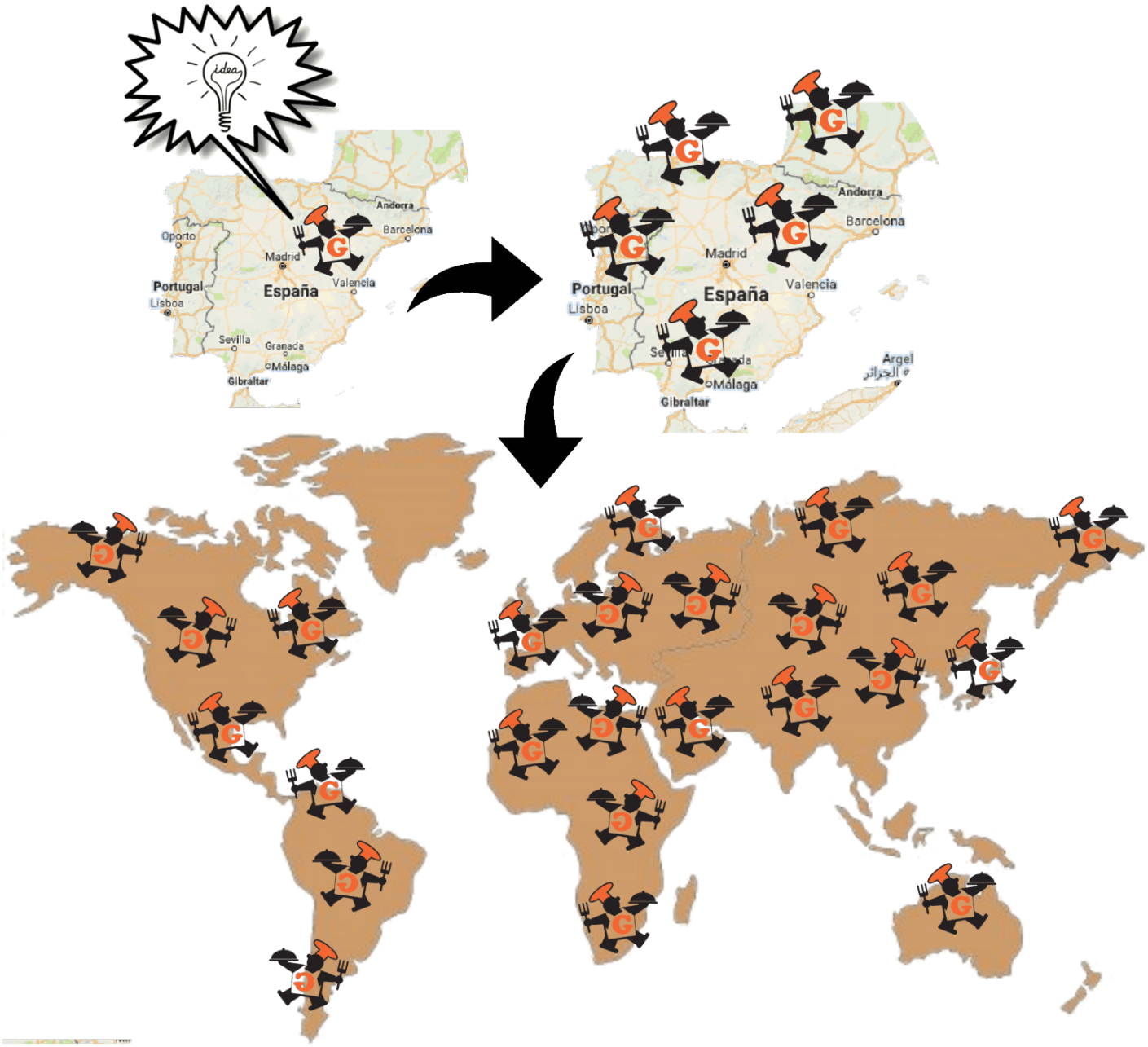


Imagen 2.2.2

¿Conocimiento Libre significa Conocimiento Gratuito?

Llegados a este punto podríamos tener la siguiente duda: Entonces, si el concocimiento libre se distribuye sin tener que abonar ningún tipo de tasa ... ¿De qué viven aquellos cuyo trabajo gira en torno a la filosofía de **conocimiento libre o abierto**? Para poder comprender este aspecto retomamos el ejemplo de **cocina abierta**. En el caso de un restaurante que siguiera la filosofía de **cocina abierta**, sus cocineros podrían obtener beneficios por los siguientes motivos:

- **Preparar los platos a clientes**. Aunque los ingredientes y los pasos de preparación son públicos siempre existe un porcentaje de usuarios que preferieren que alguien les prepare el plato de cocina, antes de ponerse a cocinar ellos.
- **Asesorar a otros restaurantes**. Otros restaurantes, advirtiendo los platos que se preparan en el restaurante de **código abierto**, estarían dispuestos a pagarle para que les asesorasen a la hora de cocinar esos mismos platos u otros similares.
- **Cursos de formación** para formar a clientes y otros cocineros en su forma de cocinar. Clientes apasionados por el tipo de cocina que se ***cuece*** en el restaurante de **código abierto** podrían asistir a cursos de formación donde se les explicaría todos los entresijos necesarios.
- **Satisfacer las demandas de clientes VIP**. Existen personas que están dispuestas a pagar al restaurante para que éste les cocine un plato a medida, un plato soñado. Este plato, una vez concebido y obtenido de él el rendimiento esperado, puede ser liberado para que otros restaurantes puedan aprovecharse de sus bondades.
- **Donativos**. Otros restaurantes y clientes, interesados en el tipo de cocina que crean los cocineros del restaurante de **código abierto**, estarían dispuestos a realizar donaciones económicas para que se siguiera **creando y liberando** nuevos platos de cocina, de los cuales se aprovecharían ellos más tarde. Si se han entendido las diferentes fuentes de financiación que podrían darse en el caso de una **cocina abierta**, estas podrían extrapolarse a cualquier otro entorno de producción de **código abierto**, entre los cuales podrían distinguirse los relativos al **software libre** (*sistemas operativos y aplicaciones libres*), y al **hardware libre** (*fabricantes de aparatos electrónicos donde se facilitan los planos y componentes para que cualquiera pueda fabricarse uno propio*).

Todo lo anterior lo podemos extrapolar al mundo del **Software Libre**, sustituyendo los restaurantes y sus cocineros por empresas de Software y sus programadores.

¿Qué otras áreas deberían ser de Conocimiento Libre?

Si se ha comprendido el ejemplo mostrado en relación a la **Cocina Libre**, ¿Qué otras áreas consideras que mejorarían el contexto mundial en beneficio de todos los individuos si fueran libres? Seguro que se te ocurren muchas, y entre esas podríamos destacar las siguientes:

- **Educación**. ¿Crees que la educación sería mejor si todos los materiales educativos disponibles fueran libres? Materiales, apuntes, exámenes, listenings, ebooks, maquetas, programas, programaciones, manualidades, experimentos, ... fueran libres. Materiales que estuvieran disponibles para poder mejorarlos, adaptarlos y volverlos a distribuir para que otros pudieran beneficiarse de ello.

Copyright 2026

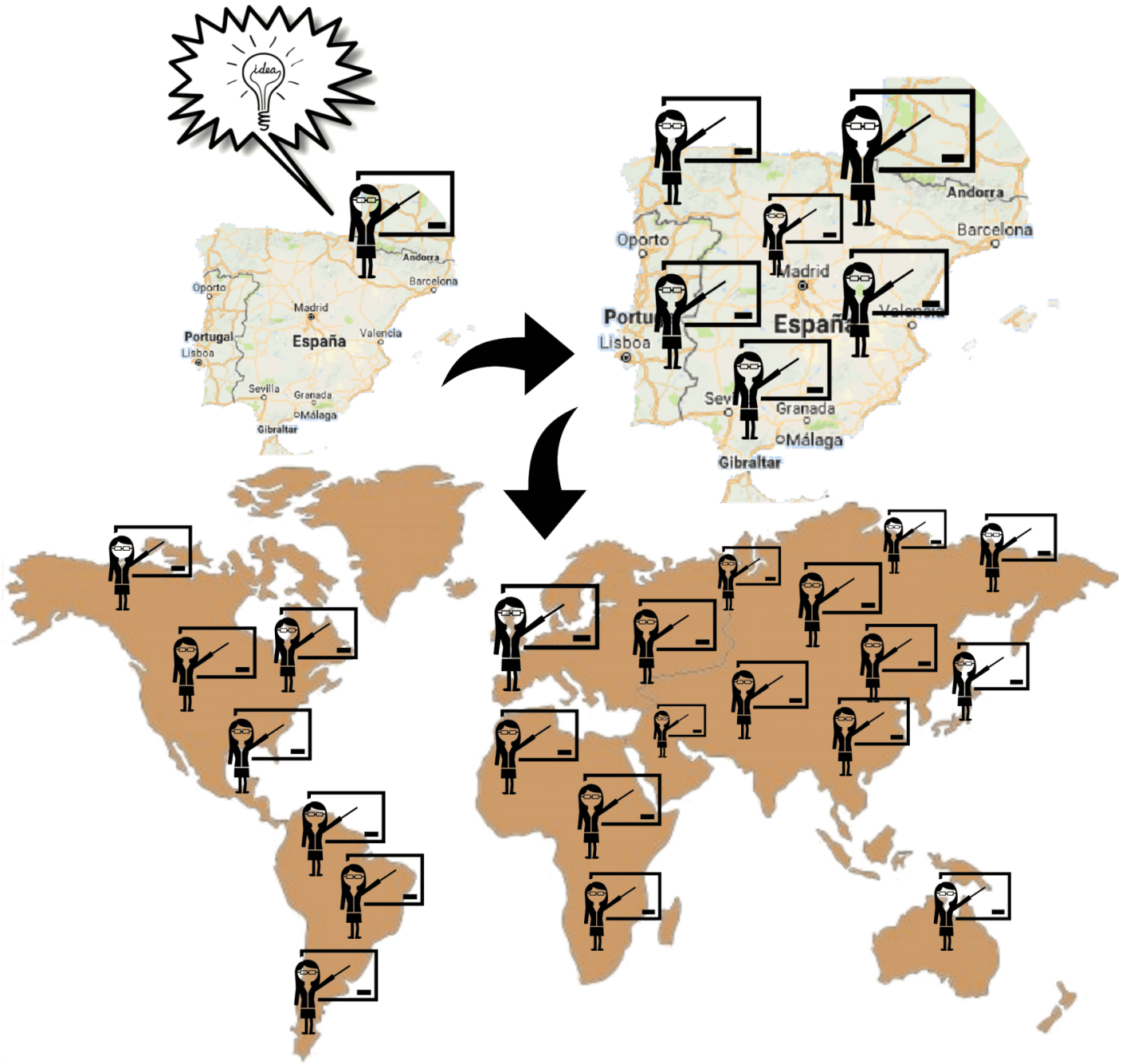


Imagen 2.2.3.1 - La cocina libre permite la distribución del conocimiento y los beneficios de manera global y en igualdad de condiciones para todos

- **Farmacia**. Si los fármacos, su composición y elaboración, fueran libres ... ¿No crees que se erradicarían las enfermedades de una forma más eficiente? El mundo farmaceútico es el claro ejemplo donde los intereses monetarios y el ansia de poder están muy por encima del bien común.
- **Software**. En una era tecnológica como la actual donde existen aplicaciones para prácticamente todo lo que se pueda uno imaginar, el software libre cobra una importancia fundamental. La posibilidad de poder compartir de manera abierta el código de un programa permite su aprovechamiento para llevar a cabo mejoras e idear nuevas aplicaciones que puedan resultar útiles. Prácticamente el 100% de los proyectos de software actuales hacen uso de código libre para acelerar su creación y crear comunidad de usuarios. Sin esta posibilidad la mayoría de los programas y aplicaciones que actualmente conocemos no existirían y los avances serían mucho menores.

Copyright 2026

Parte 1. Introducción al Software Libre

¿Qué es Software Libre?

Es muy difícil datar hechos concretos, pero se considera que el **Software Libre** nació el 27 de septiembre de 1983 gracias al ingenio del ingeniero **Richard Stallman** que por entonces trabajaba en el centro de investigación más reputado a nivel mundial, el **Massachusetts Institute of Technology** (MIT).

Richard Stallman establece las características que debe cumplir un software para considerarlo **Software Libre**. A estas premisas se las conoce como las **4 libertades del Software Libre**:

1. La libertad de **ejecutar el programa como se desee**, con cualquier propósito (libertad 0).
2. La libertad de **estudiar cómo funciona el programa**, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
3. La libertad de **redistribuir copias** para ayudar a otros (libertad 2).
4. La libertad de **distribuir copias de sus versiones modificadas** a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Por tanto, todo software que cumpla todo lo anterior lo podremos considerar **Software Libre**, lo cual guarda similitud con el ejemplo visto anteriormente de conocimiento libre referente a cocina libre.

Hitos Importantes

Entre los muchos hitos del **Software Libre** hay que nombrar y recalcar aquellos que le han dado vida y le han permitido asentar su filosofía. Estos podrían ser los siguientes:

- En 1983 **Richard Stallman** comienza a trabajar en el proyecto **GNU** (***sistema operativo libre basado en Unix***), un proyecto de colaboración para el software abierto y libre. En 1985 establece el **Manifiesto GNU** y crea la **Free Software Foundation**. Este hito es fundamental ya que **GNU** asienta las bases de lo que posteriormente daría lugar a **Linux** y este a nuestro querido **Vitalinux**. Para saber más sobre este manifiesto se recomienda dirigirse a su página web: Manifiesto GNU

- En 1985 surge del concepto o termino de licencia **copyleft** , como contrapartida al termino copyright © de derechos de autor.
- En 1989 surge la Licencia Pública General (GPL). Esta licencia declara que el código de nuestro programa cumple con las 4 libertades enunciadas anteriormente. Para saber más sobre esta licencia y otras dirigirse a la página web del proyecto GNU: Licencia de Software Libre
- En 1991 **Linus Torvalds** crea por hobby Linux. Este sería el cimiento de muchas distribuciones o sistemas operativos libres que surgirían a partir de ese momento: **Debian** , **Ubuntu**, **Red Hat**, **Mandrake/Mandriva**, **CentOS**, **ProteanOS**, etc. Algunas de estas distribuciones actualmente no son consideradas 100% libres al incluir código o programas que a su vez no lo son. Una lista más exhaustiva de distribuciones libres la podemos encontrar en la web de GNU: Lista de Distribuciones o Sistemas Operativos Libres
- El 8 de Julio de 2004 el sudafricano Mark Shuttleworth y la empresa **Canonical Ltd** anunciaron la creación de la distribución Linux más afamada que existe en el mundo: **Ubuntu**. Este hecho es fundamental en el **programa de Software Libre de la DGA Vitalinux** ya que éste se basa en el sistema operativo **LUbuntu** (***Light Ubuntu****), **una versión ligera de** Ubuntu**.

Copyright 2026

Parte 2. Distribuciones Libres: Vitalinux

Copyright 2026

Presentación Contenidos Parte 2

Dentro del mundo del **Software Libre** cabría distinguir entre **aplicaciones libres** y **distribuciones o sistemas operativos libres**. La diferencia fundamental radica en que un sistema operativo es el software base que se instala sobre la máquina física, haciendo de intermediario entre la máquina y el usuario, mientras que las aplicaciones son pequeños trozos de software que se instalan sobre el sistema operativo proporcionandole funcionalidades. En esta parte del curso se va a presentar la distribución o sistema operativo libre de la DGA: **Vitalinux**.

Para comprender mejor esta parte, es muy importante haber mirado y comprendido la parte anterior del curso referente al concepto de **Conocimiento y Software Libre**, ya que en ella se presenta una terminología que aquí se dará por supuesta: **Software Libre**, **GNU**, **Linux**, etc.

Para comprender a la perfección esta parte del curso sería recomendable descargar, instalar y probar el sistema operativo **Vitalinux**. Esto no es un requisito para poder superar el curso, pero es de entender que es la mejor forma de conocerlo, probándolo.

Copyright 2026

Parte 2. Distribuciones Libres: Vitalinux

¿Qué es Vitalinux?

Vitalinux es un sistema operativo similar a *Microsoft Windows* o *Android* que nos va a permitir trabajar con nuestro equipo informático (*pc sobremesa, portatil, tablet o dispositivo móvil*) con las mismas aplicaciones o similares a las que normalmente utilizamos (*Mozilla Firefox, Google Chrome, LibreOffice, etc.*), pero con la gran diferencia de que el **software es libre**, de **código abierto**. En concreto, **Vitalinux**, al igual que **Android**, no es un sistema operativo que nace desde cero, sino que esta basado y tiene sus raíces en otro sistema operativo llamado **GNU/Linux**. Más concretamente, **Vitalinux** esta basado en la **versión ligera del sistema operativo GNU/Linux Ubuntu**, del cuál podríamos destacar entre otras muchas carácterísticas estas tres:

- Es un sistema operativo libre
- Al basarse en la versión ligera de Ubuntu llamada Lubuntu se garantiza que el tiempo de respuesta del equipo sea el menor posible
- Incorpora un cliente Migasfree

A continuación se detallan cada una de estas caracteríasticas

1. Vitalinux es un sistema operativo libre de Código Abierto

Al igual que cualquier otro sistema operativo GNU/Linux (*Debian, Ubuntu, OpenSuse, ...*), **Vitalinux es un sistema operativo libre**. Esto significa que el código bajo el cual esta programado es **abierto** y por tanto cualquiera puede copiarlo, modificarlo, mejorarlo y distribuir el resultado.

Para poder comprender esta primera gran característica del sistema Vitalinux podríamos indicar que gracias a que Ubuntu sigue esta filosofía de código abierto, ha permitido que desde el Departamento de Educación se haya podido aprovechar todo el enorme trabajo que hay tras el sistema operativo Ubuntu, adaptarlo a las necesidades demandadas por los centros educativos, y poderlo distribuir posteriormente bajo el nombre de Vitalinux. De otra forma, **si hubiera sido necesario crear un sistema operativo desde cero, hubiera sido una tarea imposible**.

A diferencia de otros sistemas operativos privativos (*p.e. Microsoft Windows*), gracias a esta característica, **sin incurrir en ninguna ilegalidad** se ha podido eliminar todo aquello que entendíamos que era prescindible de Ubuntu y al mismo tiempo añadir todo lo que echamos en falta, para posteriormente distribuir el resultado para adecuarlo a las demandas de los centros educativos.

Copyright 2026

2. Vitalinux está basado en un sistema operativo muy ligero

Al basarse en la versión ligera de Ubuntu llamada **Lubuntu** se garantiza que el **tiempo de respuesta** del equipo sea **el menor posible**. Es decir, permite que equipos antiguos con pocos recursos hardware puedan ser funcionales, ya que el sistema base (*sin arrancar ninguna aplicación de Escritorio*) tan sólo requiere unos 200MB de memoria RAM. De esta forma, si lo miramos desde el punto de vista opuesto, si tenemos un equipo actual (*varios Gigas de memoria RAM, un procesador de última generación, etc.*) el uso de un sistema operativo de estas características hará que el equipo "vuele".

No obstante, esto no puede llevarnos a engaños ni confusiones ... hay que aclarar que aunque el sistema operativo base sea muy liviano, **si las aplicaciones que usamos son pesadas, un equipo con bajas prestaciones podrá tener penalización**. Un claro ejemplo podría ser el navegador Web Google Chrome, que en este tipo de sistemas ligeros suele consumir muchos más recursos de RAM y procesador que el propio sistema operativo bajo el cual esta funcionando dicha aplicación.

Es decir, **Vitalinux** es capaz de arrancar y permitir trabajar a un usuario con un equipo de hace 10 o 15 años posibilitando su reutilización, pero de poco sirve que el sistema operativo sea ligero si las aplicaciones que se instalan sobre él no lo son. Equipos del programa de la **Escuela 2.0** que fueron repartidos entre los centros educativos de la Comunidad Autónoma de Aragón (*p.e. Toshiba M200/400/700, Toshiba NB250/510, HP TC4400, etc.*) disponen de serie entre 512MB y 1GB de memoria RAM, suficiente para que se pueda instalar e iniciar **Vitalinux**, pero con poco margen restante para el resto de aplicaciones. Hay que tener en cuenta que actualmente en la compra de un ordenador se recomienda un mínimo de 4GB de RAM (*Windows 10 recomienda para empezar a funcionar 2000MB=2GB*), muy por encima de la disponible por estos equipos. Por este motivo, con la finalidad de que el usuario pueda lanzar simultáneamente varias aplicaciones (*se cargan en la memoria RAM junto al sistema operativo*) en este tipo de equipos es recomendable ampliar la memoria RAM hasta 2GB.

3. Vitalinux incorpora un cliente Migasfree

Aunque existen otras muchas características que podríamos seguir destacando del sistema operativo Vitalinux, algunas de las cuales las iremos apreciando a lo largo de la realización del curso, por no alargar este primer punto introductorio cabría señalar y destacar **aquello que le hace diferente a Vitalinux del resto** de distribuciones Linux que existen en el mundo (*p.e. Ubuntu, Red Hat, Suse, etc.*) y de Windows: tiene preinstalado un **cliente Migasfree**.

Este cliente Migasfree nos permite entre otras muchas cosas el poder **gestionar el software del equipo Vitalinux de manera remota, desatendida y automatizada**, e **inventariar todo su**

software y hardware. Para que nos hagamos una idea, Migasfree nos permite a demanda del administrador de los equipos del centro, y sin la necesidad de la intervención de ningún usuario cosas como las siguientes:

- Permite **programar simultáneamente el apagado automático de todos los equipos del centro** a una determinada hora. Esto evita que queden equipos encendidos de manera ociosa fuera del periodo escolar.
- Permite **instalar simultáneamente en todos o en un conjunto de equipos del centro un conjunto de programas**. Esto evita tener que ir uno a uno instalando cada uno de los programas informáticos que pueden necesitarse en la impartición de determinadas materias.
- Permite **desinstalar programas o borrar ficheros** simultáneamente en todos o en un conjunto de equipos del centro que se consideren innecesarios o perjudiciales para el equipo informático. Gracias a ello, a través de **Migasfree** sería posible hacer una limpieza automatizada en todos los equipos de un centro educativo, a principio o final de curso, eliminando todo aquello que no nos interese mantener, provocando una normalización en los equipos.
- Permite **mantener actualizados** todos los equipos del centro educativo. Es decir, en el caso de detectar un error cualquiera del sistema o una posible mejora, su solución puede desplegarse a través de **Migasfree** en todos los equipos del centro.
- Nos permite **disponer de un inventariado muy detallado de todos los equipos del centro**, tanto de los componentes hardware que componen los equipos, como de todo el software que hay instalado en ellos. Resaltar en este aspecto que este inventariado se hace de manera transparente y desatendida para los usuarios, siendo **Migasfree** el encargado de recolectar todos los datos del equipo y de registrarlos en una base de datos que posteriormente puede consultarse.
- ... Todo aquello que se nos ocurra que tenga que ver con el software de los equipos informáticos (*personalizar el fondo de Escritorio de manera desatendida, configurar acceso a libros digitales de las diferentes editoriales, configurar impresoras o fotocopiadoras, etc*).

Como puede apreciarse, todo ello redunda en un desahogo del administrador de los equipos informáticos garantizando una mayor fiabilidad y eficiencia de todos los sistemas informáticos del Centro Educativo

> **¿Quiénes son los creadores y desarrolladores de Migasfree?** Destacar por último que sorprendentemente Migasfree es un software creado, desarrollado y liberado por dos trabajadores del Ayuntamiento de Zaragoza (España): **Alberto Gacías y José Antonio Chavarría**

Copyright 2026

Parte 2. Distribuciones Libres: Vitalinux

Wiki de Vitalinux

El proyecto de Software Libre **Vitalinux** dispone de una Wiki de referencia en la cual se públican todos las novedades, noticias, avisos, contenidos (*cursos de formación*), FAQs, ... y demás aspectos que le conciernen.

Entre toda la información que allí se expone, cabría destacar la siguiente:

- **Área de Descargas**: Desde allí podremos descargar las últimas versiones disponibles en formato ISO de Vitalinux.
- **Noticias**: Se publican las últimas noticias referentes a Vitalinux y las aplicaciones que se ejecutan sobre él. Estas están intimamente relacionadas con la Web de soporte, ya que muchas de ellas trantan de informar sobre el estado de alguna incidencia que esta generalizada a todos los centros educativos que hacen uso de Vitalinux.
- **Mapa de Centros Vitalinux**: Mapa actualizado de las localidades de la Comunidad Autónoma de Aragón y centros educativos que hacen uso de Vitalimux de manera oficial.
- **FAQs**: Preguntas frecuentes y documentación de interés general.
- **Robótica**: Información sobre asuntos de robótica en Vitalinux. Esta información es generada a partir de incidencias abiertas por los centros educativos a este respecto.
- **Libros digitales o eBooks**: Soporte que damos desde Vitalinux a los libros digitales distribuidos por las diferentes editoriales usadas en los centros educativos.
- **Pizarras Digitales Interactivas o PDIs**: Información de la compatibilidad existente entre Vitalinux y los diferentes modelos de PDIs que hay distribuidos por los centros educativos.

Además de la Wiki los responsables del programa Vitalinux también están trabajando en el desarrollo de una Web de documentación técnica que la suplirá en un futuro y donde se podrá encontrar de una forma más automática e intuitiva todo lo referente al programa de Software Libre de la DGA.

Copyright 2026



Si además de la documentación necesitas algo mas de información o ayuda de forma gráfica, puedes consultar el canal de youtube con varios vídeos sobre vitalinux y sobre todo con las Píldoras Formativas

Parte 2. Distribuciones Libres: Vitalinux

Web de Soporte Vitalinux

Con la finalidad de dar soporte técnico a todos los centros educativos de la Comunidad Autónoma de Aragón que hacen uso de **Vitalinux** existe una web donde previo registro un usuario puede solicitar ayuda y asesoramiento sobre cualquier aspecto concerniente a **Vitalinux**: Web de soporte

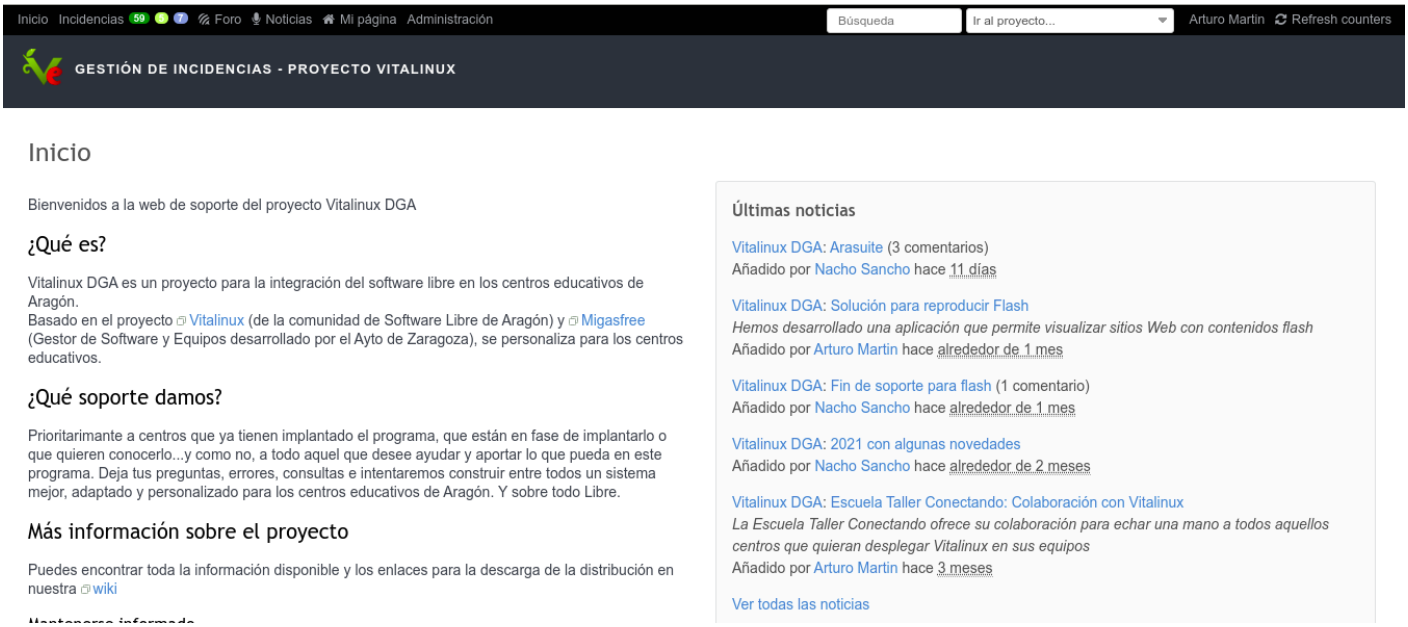


imagen 3.4.1 - Web de Soporte del Programa Vitalinux

Los usuarios que actualmente están registrados en dicha Web de soporte suelen corresponderse con los coordinadores Vitalinux de los centros educativos, pero pueden registrarse todos aquellos usuarios interesados en Vitalinux y todo su ecosistema. Estos coordinadores son las personas que han sido designadas en sus centros para encargarse de la gestión de los equipos Vitalinux, haciendo de intermediarios entre el centro educativo y los técnicos informáticos de Vitalinux. A continuación se detallan los posibilidades que esta web de soporte le ofrece al coordinador Vitalinux o al usuario que esté registrado:

- **Abrir una nueva petición o incidencia**. Teniendo en cuenta que Vitalinux es un sistema que se gestiona de manera remota y desatendida mediante Migasfree, los coordiandores Vitalinux usan esta vía para solicitar cualquier cosa relacionada con Vitalinux: instalación/desinstalación de programas para los equipos de su centro, configuración de nuevas impresoras, personalización de su entorno de Escritorio, gestión

de cuentas de usuario, etc. Al ser un programa totalmente abierto, pueden consultarse públicamente todas estas peticiones, a excepción de aquellas que incluyan datos privados (*contraseñas, datos relevante de alumnos, etc.*): Lista de peticiones de la Web de Soporte

> **Abrir incidencias de forma Anónima**: es posible añadir una nueva petición o incidencia de forma anónima, sin necesidad de registro previo, indicando como destinatario **soportevitalinux@educa.aragon.es**.

- **Noticias**: Desde la Web de soporte se publican todas las noticias referentes al programa, que posteriormente son enlazadas en la Wiki del proyecto. Estas noticias pueden ser publicadas por los usuarios registrados, por lo que suelen corresponderse con noticias de perfil técnico o divulgativo.
- **Foros**: Los foros se corresponden con debates, discusiones, dudas o cualquier otro asunto relacionado con Vitalinux en los que se invita a participar a la comunidad de usuarios registrados. Estos foros, al igual que las noticias, son de acceso público como puede comprobarse, pero únicamente pueden partipara usuarios registrados.

Copyright 2026

Parte 2. Distribuciones Libres: Vitalinux

Donde descargar Vitalinux

Al tratarse de un Sistema Operativo, la forma más habitual para distribuir/descargar Vitalinux es usando el **formato ISO**. A continuación veremos donde encontrar la versión más actualizada para su descarga.

¿Qué es un archivo o imagen ISO?

La extensión ***.iso*** en un archivo nos advierte de que ese archivo es el resultado de comprimir y empaquetar un conjunto de archivos y directorios. A modo de ejemplo existen muchos programas que nos permiten crear un archivo o imagen ISO a partir de un CD de música o DVD de vídeo, obteniendo como resultado un único archivo resultante **.iso**, el cual contiene todos los archivos existentes en el CD o DVD, posibilitando posteriormente su copia o restauración.

En lo que a este curso respecta, en el ámbito de los sistemas operativos, una imagen ISO de un sistema operativo (*p.e. Vitalinux*) hace referencia a un único archivo que aglutina en su interior a todos los archivos necesarios, que volcados sobre un disco duro, o pendrive, o DVD, permiten a un equipo informático arrancar bajo ese sistema, interactuar y trabajar con él.

Área de DESCARGA de ISOS e Imágenes

Como ya se ha explicado previamente el programa de Software Libre **Vitalinux** dispone de una Wiki de referencia en la cual se públican todos las novedades, noticias, avisos, contenidos (*cursos de formación*), FAQs, ... y demás aspectos que le conciernen. En concreto, en esta Wiki podemos encontrar el Área de Descargas desde la cual obtener las imágenes ISOs para su posterior prueba e instalación.

Una vez hayamos descargado la imagen ISO de Vitalinux, antes de hacer uso de ella, es aconsejable comprobar la integridad del archivo descargado, ya que al tratarse de archivo muy pesados (*tamaño > 2GB*) es posible que durante el proceso de descarga se haya producido algún error.

Desde élÁrea de Descargas podrás descargar el sistema Vitalinux EDU DGA en sus diferentes formatos, siendo estos:

Imágenes ISO de Vitalinux 2.0 . Se trata de la última versión y siempre **será la preferente**. Tras descargar estas imágenes ISOS podrás crear un **DVD o USB arrancable** que te permitirá probar Vitalinux en modo Live o directamente instalar el sistema en un equipo informático.

¿64 o 32 bits?. Deberás seleccionar la arquitectura de tu ordenador. Por norma general, y si no lo sabes, será de **64 bits**. Si tu ordenador no lo soporta, el propio instalador te lo dirá

Imágenes ISO de Vitalinux 1.0 . Se trata de la versión antigua. Por defecto no se recomienda y solo será necesaria en casos muy concretos: problemas con ciertos periféricos o hardware muy antiguos y específicos, la versión 2.0 funciona en equipos del siglo pasado.

¿Es necesario descargar ésta ISO?. Si no contamos con un ordenador con **Vitalinux** ya instalado (por ejemplo: nos ha facilitado uno desde el centro educativo), necesitaremos la ISO para instalarla o bien en una máquina virtual o en una máquina física como veremos en la Parte 2.

Cualquiera de los archivos anteriores son de **un gran tamaño** (*>2GB*) al aglutinar en un único archivo al sistema operativo **Vitalinux** con todas sus aplicaciones ya preinstaladas (*herramientas ofimáticas, programas multimedia, utilidades, etc.*).

Por ello, cuando nos descargamos un fichero de Internet, y en éste caso un fichero tan grande, es importante **verificar que el fichero se ha descargado correctamente** (*verificar su integridad*). Ocurre más a menudo de lo que nos gustaría, que la descarga no se realiza de forma correcta y luego perdemos mucho tiempo hasta que nos damos cuenta de que el error está en una descarga errónea o fallida.

Comprobar la Integridad de un fichero descargado

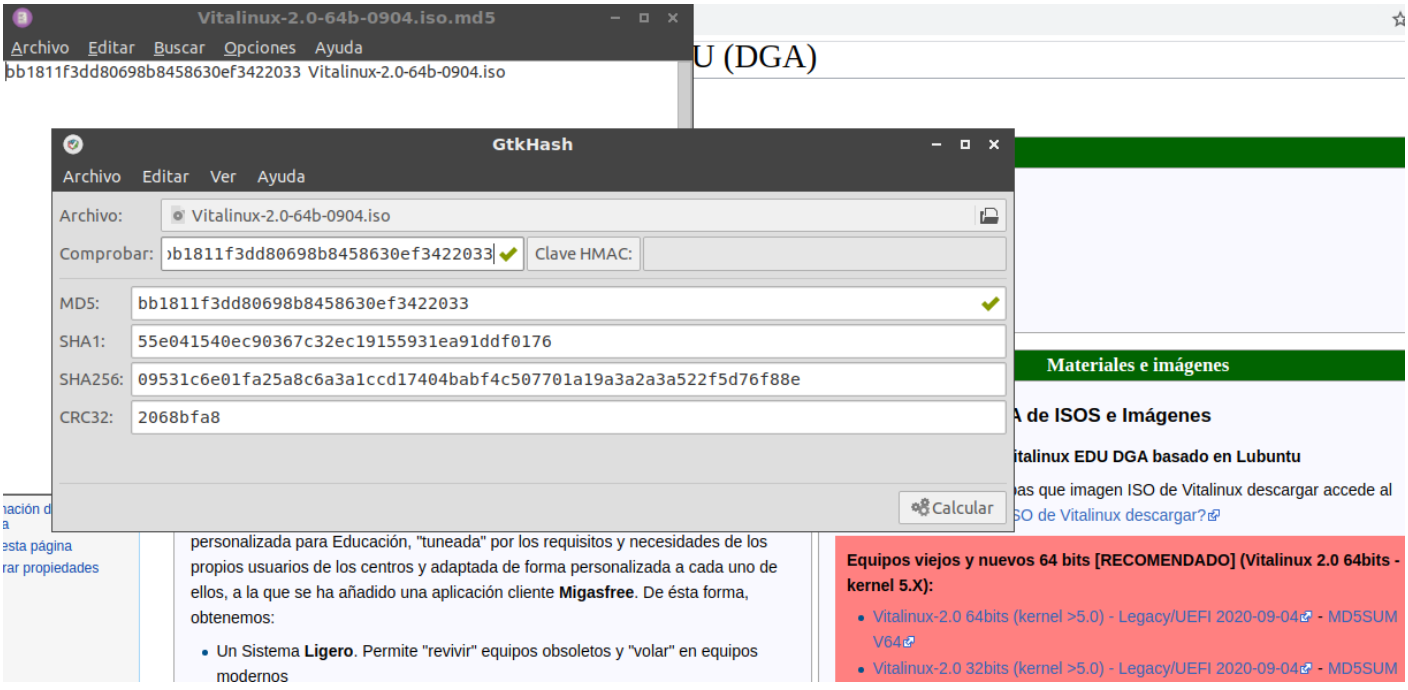
Para verificar que la descarga de un fichero ha sido correcta existen las "*firmas*", "*resúmenes*" o "*ficheros hash*". Si observas, en el Área de Descargas al lado del fichero a descargar tienes otra descarga disponible: **MD5SSUM**. Prueba a bajar uno y abrirlo con el bloc de notas o cualquier editor de texto plano.

Observarás que es un fichero de texto que contiene simplemente una línea con el **resumen** del fichero y el nombre del fichero. El resumen de un archivo es una cadena de texto de tamaño fijo (*32 caracteres*) resultante de aplicar un algoritmo al fichero original, de forma que si el archivo original cambiara en lo más mínimo, el resultado de aplicar de nuevo el resumen sería completamente distinto.

Así pues, si aplico el algoritmo (*MD5 en éste caso*) al fichero ***.iso** que me he descargado y resulta la misma cadena que contiene en su adjunto ***.iso.md5**, puedo asegurar que la descarga se realizó con éxito.

¿Y cómo hago ésto? Dependerá del Sistema Operativo que uses para descargar y comprobar los ficheros de descargas:

- Si usamos una **distribución GNU/Linux**, puedes hacer la comprobación mediante el programa gráfico **gtkhash**, o directamente desde una terminal haciendo uso del ejecutable llamado **md5sum**.



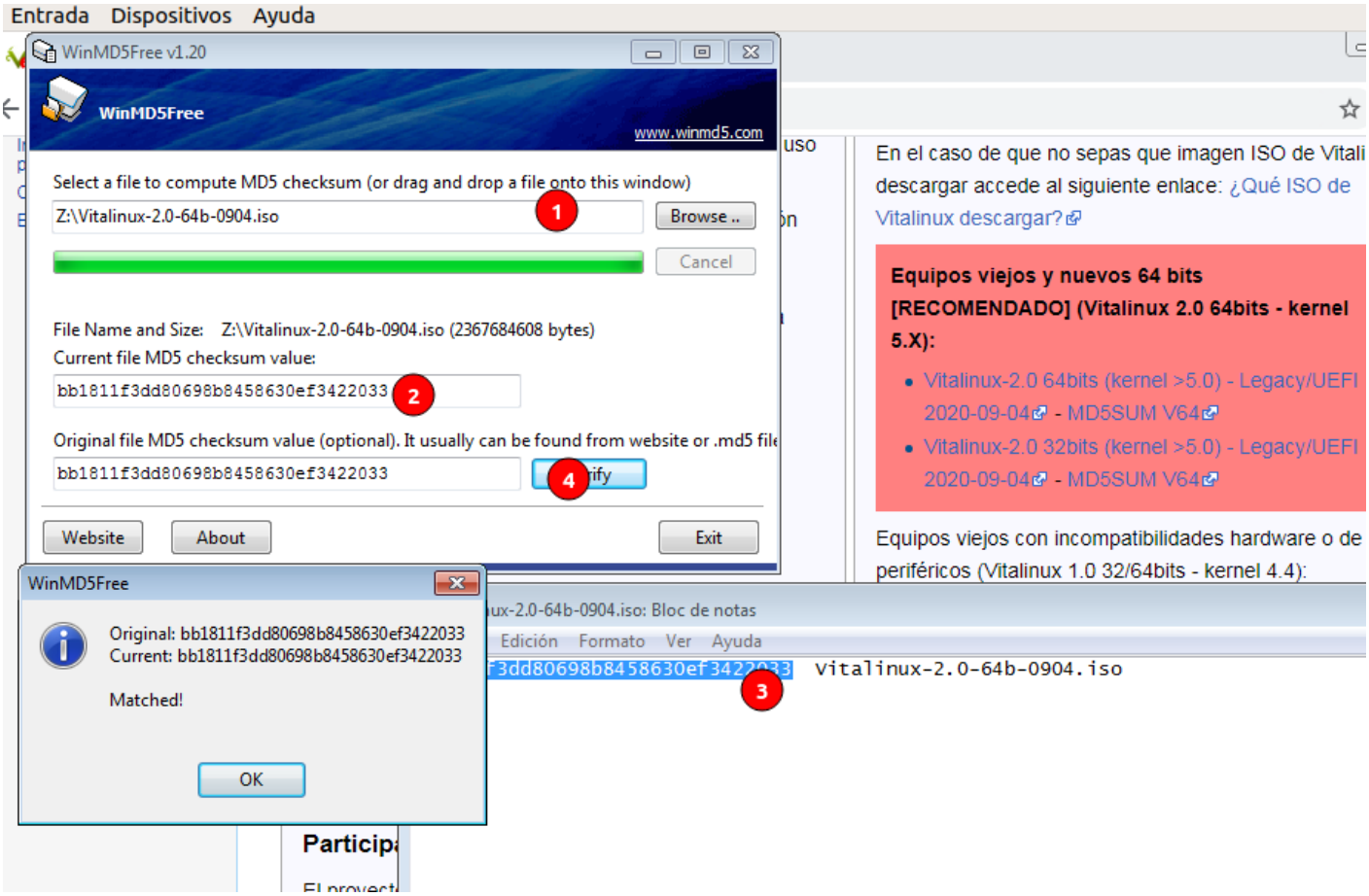
1. Mediante **gtkhash**. Instala y lanza **gtkhash**. Una vez abierto el **gtkhash** selecciona el archivo (*p.e. *.iso*) del cual quieres comprobar su integridad y pincha en **calcular Hash**, habiendo pegado en el campo **comprobar** el código que debería obtenerse (*p.e.*

*contenido del archivo *.iso.md5*).

2. Mediante el ejecutable **md5sum**. Abre una terminal (CONTROL+ALT+T) y navega hasta el directorio donde se localiza el archivo (*p.e. *.iso*) del cual quieres comprobar su integridad, junto al fichero md5 (*p.e. *.iso.md5*) y ejecutas (*si se verifica el fichero saldrá un resultado de "**La suma coincide**"*):

```
cd Descargas
md5sum -c fichero_md5_descargado.md5
```

- Si usamos **Windows**, podemos descargarnos el programa winmd5free. Lo ejecutamos y seguiremos unos sencillos pasos:
  1. Ejecuta el programa y busca el archivo del cual deseas comprobar su integridad (*p.e. *.iso*)
  2. El programa comprobará el fichero y generará un **resumen** o **hash** resultante
  3. Abre con un **Bloc de Notas** el fichero con extensión MD5 que descargaste que contiene el **resumen MD5** del archivo original (*p.e. *.iso.md5*).
  4. Comprueba que el resumen generado por el programa a partir del archivo, y el resumen proporcionado por el archivo MD5 son iguales. Si la firma coincide podemos concluir que está todo correcto, que el archivo descargado está integro.



Copyright 2026

Parte 2. Distribuciones Libres: Vitalinux

Como probar Vitalinux

La pretensión de esta parte del curso **NO es enseñar a instalar** un sistema operativo en un equipo, sino conocer que es **Vitalinux** y su cliente **Migasfree**, pero creemos que es necesario añadir este apartado informátivo para saber como hacerlo.

> ¿¿Necesito Probar Vitalinux?? Recordamos que no es necesario descargar e instalar Vitalinux en un equipo para aprobar este curso, pero es recomendable hacerlo para entender y comprender las bondades de este sistema operativo.

Aunque la opción más sugerente es instalar **Vitalinux** en un equipo físico, esta no es la opción más sencilla, ni las más recomendada para tener un primer contacto con un nuevo sistema operativo. Para estos menesteres disponemos de **Virtualbox**, el cual se presenta en el siguiente apartado.

Instalación de Vitalinux en Virtualbox

En este apartado del curso se va a presentar a el afamado y potente software **VirtualBox** que nos va a permitir poder probar y testear el sistema operativo que deseemos, además de poder instalar sobre él las aplicaciones que nos interesen pudiendo trabajar con ellas de igual forma a como lo haríamos si las tuvieramos instaladas en nuestro equipo físico.

Virtualbox es la herramienta ideal para la realización de pruebas y test de sistemas operativos. Se trata de una aplicación software de Escritorio que nos permite instalar y probar sistemas operativos sin afectar en absoluto al sistema operativo que tenga instalado el equipo físico.

Tal cómo lo definen en su página oficial "***VirtualBox*** *es un poderoso software de virtualización tanto para la empresa, como para el uso doméstico. Además se caracteriza por ser la única solución profesional que está libremente disponible como software de código abierto bajo los términos de la Licencia Pública General de GNU (GPL v2)*".

En definitiva, **VirtualBox** es un software muy interesante que nos va a permitir crear una máquina virtual, para posteriormente sobre ésta instalar y probar un sistema operativo (*p.e. Vitalinux*) y todas sus aplicaciones obteniendo como resultado exactamente lo mismo que si lo hubiéramos hecho directamente sobre el equipo físicamente.

Copyright 2026

{% notificacion_important title='¿Qué significa que la máquina es Virtual?' %} **Virtualbox** nos va a permitir crear máquinas virtuales en un sentido metáforico, ya que cuando creamos una máquina en Virtualbox en realidad estamos cediendo parte de los recursos hardware de la máquina física a la máquina creada. Es decir, a modo de ejemplo, si disponemos de un equipo físico con 4GB de memoria RAM y creamos una máquina en Virtualbox con 1GB de memoria RAM, ese GigaByte es real (*no es virtual*) ya que se los esta **quitando a la máquina física** dejándola únicamente con 3GB. Entendido lo que sucede con la memoria RAM de la máquina virtual, exáctamente igual podríamos decir de la CPU, la tarjeta de sonido, las tarjetas de red, etc ... Por tanto, Virtualbox es un software que tiene la capacidad de hacernos creer que tenemos varias máquinas en una. {% endnotificacion_important %}

Crear una máquina virtual es tan facil como seleccionar la acción de Nueva Máquina Virtual, y seguir los pasos que se indican en el asistente. Si dejamos todo por defecto no tendremos problemas, pero por revisar:

- **Nombre y Tipo de máquina**. Nombre deseado y tipo Linux (32 o 64, según deseemos)
- Memoria **RAM** que le asignamos (1GB por ejemplo está bien)
- Crear un **disco** Virtual (se recomienda cuando lo pida reservar el espacio dinámicamente para que solo ocupe en disco el espacio que gastemos). El tamaño del disco debe ser **superior a 25 GB** para no tener problemas de espacio

Una vez creada la máquina virtual, tendremos que ir a su configuración e indicar la ruta de la ISO de Vitalinux para posteriormente ya arrancarla.

Con la finalidad de tratar que sea más comprensible el **proceso de creación de una máquina Virtual en Virtualbox** se ha creado el siguiente videotutorial (*advertir que este videotutorial se realizó para el curso de Aularagon de Vitalinux, por lo que las referencias a dicho curso habrá que omitirlas):*

https://www.youtube.com/embed/NsVrwUFeYrc

Parte 3. Aplicaciones Libres

Copyright 2026

Parte 3. Aplicaciones Libres

Introducción a las Aplicaciones Libres

Dentro del ecosistema del Software en general, nos encontramos con la parte más visible del mismo: las aplicaciones. Las aplicaciones son los programas informáticos con los que interactuamos con el ordenador (o dispositivo) y son, al fin y al cabo, la esencia del uso o necesidad de las nuevas tecnologías.



Nos van a facilitar la realización de tareas complejas. A diferencia del sistema operativo, cuyo objetivo es gobernar el hardware y poner orden en la ejecución de los diferentes programas, las aplicaciones realizan funciones específicas. Gracias a ellas, podemos disponer de un centro ofimático, visualizar y editar imágenes, navegar por internet, hacer cálculos matemáticos complejos (o sencillos...). Existen aplicaciones sencillas o auténticas herramientas/suites complejas realizadas con miles o millones de líneas de código

Copyright 2026

Las aplicaciones no dejan de ser un **programa escrito en un lenguaje de programación** que realiza unas funcionalidades deseadas, pero podremos disponer de Apliacaciones Libres o Aplicaciones No Libres, Comerciales o No comerciales, de código abierto pero no libres (con una licencia restrictiva por ejemplo), gratuitas o de pago, y además compiladas (o disponibles) para ciertos sistemas operativos (y otros no) o incluso para solo algunas arquitecturas concretas (por ejemplo 64 bits)

En nuestro caso nos interesa profundizar en las aplicaciones libres. Las **razones**, tal como hemos hablado al principio son aplastantes en contraposición a las aplicaciones comerciales, y más en el ámbito de la educación:

- El código es **auditable**, garantía de que el código hace lo que tiene que hacer y no realiza acciones en segundo plano. Podemos modificar el código. Al tener disponible el código y poder modificarlo, podemos cambiar algo que no nos guste, personalizarlo e incluso involucrar a los alumnos en dicho proceso
- Podemos distribuir el software de forma **ética** entre nuestros alumnos. El alumno puede disponer de dicho software en casa sin necesidad de "piratear" o realizar alguna inversión.
- Como norma general, el software libre está **disponible** para todas las plataformas. Por tanto, no dependemos de usar un sistema operativo concreto para poder hacer uso del mismo. Qué ocurre si el alumno tiene en casa un Linux y en el centro un Windows? O al revés? O usa MacOs?
- La excusa de que es el software que se usa **en la empresa** solo puede llegar a ser justificable en etapas de Formación Profesional (e incluso en éste caso podría llega a ser discutible). El objetivo es que el alumno adquiera competencias tecnológicas, y las aplicaciones libres permiten dicha adquisición sin problemas
- "*Las aplicaciones propietarias funcionan mejor que las libres...*". Es completamente discutible e incluso falso en muchas ocasiones. Tendemos a confundir desconocer una apliación con que **funcione** mal. Hay ejemplos de aplicaciones libres muy consolidadas y con un funcionamiento ejemplar. Libreoffice, Gimp o Firefox son suites y aplicaciones libres Incluso la nave (Ingenuity) que llevó al Robot (Perseverance) a Marte, estaba basado en Software Libre Open Source, y el código se puede consultar y rehutilizar

Terminamos ésta primera parte con una lectura necesaria sobre porqué una institución educativa debería usar y enseñar software libre por el proyecto GNU:

1. Permite **compartir** conocimiento y herramientas
2. **Responsabilidad Social** ante el posible dominio de un desarrollador o multinacionales
3. **Independencia** del producto
4. **Aprendizaje** de la propia herramienta, la ética y la práctica profesional
5. **Ahorro de costes** que puede redundar en otras áreas con necesidad
6. **Calidad**

Veamos pues, qué aplicaciones y cómo tenerlas disponibles

Copyright 2026

Parte 3. Aplicaciones Libres

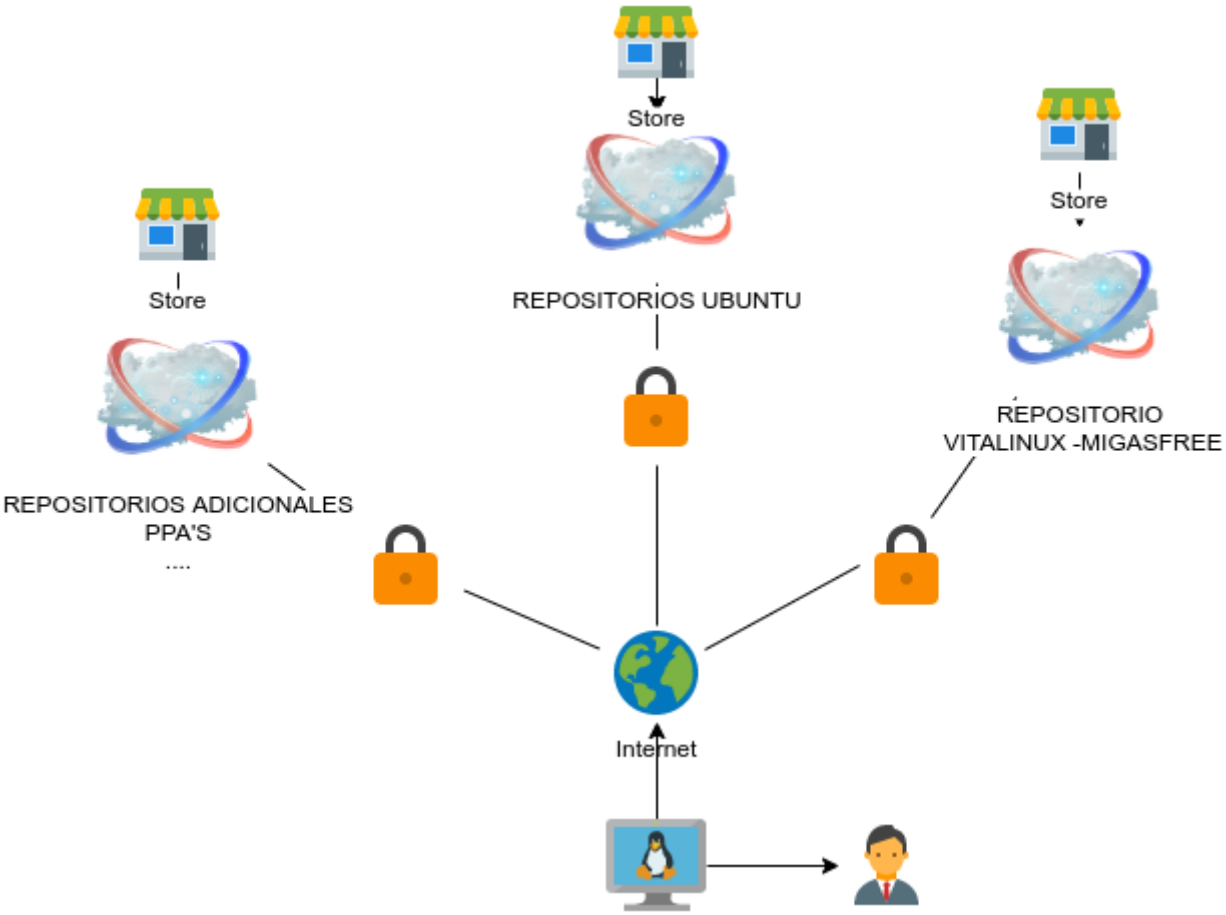
De dónde viene el Software

Una de las claves para el uso de las aplicaciones es la **distribución**. Recordamos que una de las libertades del software libre es que puedes redistribuir copias para ayudar a otros o incluso puedes distribuir copias modificadas. Es decir, que podemos coger cualquier programa hecho bajo la licencia de software libre, y además de ser libres para poderlo modificar y mejorar, podemos posteriormente entregarlo a quien queramos y de la forma que queramos.

Repositorios

Ahora bien...cómo lo distribuimos? Existen múltiples canales y gracias a esta libertad de distribución es posible aglutinar todo el software libre en determinadas ubicaciones de Internet y distribuirlo a quien lo desee de una manera centralizada. Estas ubicaciones se denominan en el mundo linux como **Repositorios**.

Copyright 2026



Un Repositorio es una especie de almacen/tienda, donde puedes consultar las aplicaciones disponibles solicitar su descarga e instalación. Para hacernos una idea del concepto, android (que también es un Linux) tiene preconfigurados unos repositorios de Google de tal forma que cuando nosotros buscamos e instalamos una nueva aplicación a través de su "Play Store", lo estamos haciendo entre el software recolectado por dichos repositorios. En el mundo Linux además contamos con las siguientes ventajas:

1. Existen **dependencias** entre aplicaciones. Muchas veces, una aplicación necesita de una o varias aplicaciones adicionales para funcionar (por ejemplo librerías). Gracias a éste mecanismo, cuando solicitemos instalar una aplicación se buscarán e instalarán las aplicaciones necesarias. Ésto aporta la ventaja, de que si una misma librería va a ser usada por varias aplicaciones...¿para que "engordar" cada aplicación con la misma librería? Simplemente se marca la dependencia, y si no la tiene (de alguna necesidad previa, la descargará e instalará por nosotros
2. **Actualización**. Los repositorios mantendrán actualizadas las aplicaciones. Es decir, si se encuentra un error o una mejora y se debe actualizar la aplicación, se actualizará en dicho repositorio, y nuestro sistema actualizará dicha aplicación porque la encontrará en una versión superior a la que tiene instalada.

3. **Seguridad**. Los repositorios vienen con un sistema de autenticación basado en firmas, de forma que establecemos una comunicación segura y los paquetes vienen firmados. Es decir, si confiamos en dicho repositorio, se nos proporcionarán las claves para verificar que estamos instalando el software de dicho repositorio, y no de otro.

Repositorios preconfigurados

Por defecto, en Vitalinux están configurados los siguentes repositorios

- Repositorios oficiales *de ubuntu**. Gran parte del software instalado proviene de la distribución de ubuntu base que lleva Vitalinux
- Vitalinux 1.0 basado en Ubuntu Trusty
- Vitalinux 2.0 basado en Ubuntu Bionic
- Respositorios de **Libreoffice**, para mantener una versión de la suite ofimática más actualizada que la que viene por defecto en Ubuntu. Así, contamos con las siguientes versiones:
- Vitalinux 1.0 con Libreoffice 6.1
- Vitalinux 2.0 con Libreoffice 6.4 (en breve pasaremos a la 7.0)
- Repositorios de **aplicaciones comunes**, que no vienen de forma oficial en Ubuntu: Google Chrome, openjdk
- Repositorio **propio de migasfree**. En éste tenemos disponibles muchas **aplicaciones propias del proyecto** para personalizar la distribución, así como **aplicaiones de terceros** que dejamos disponibles de forma actualizada para no tener que añadir repositorios de forma manual. Por ejemplo: obs-studio, scratch, geogebra... y muchas más que veremos en capitulos posteiores.

* **-** *Los repositorios de Ubuntu están ("congelados") en un servidor del proyecto, para controlar así las actualizaciones periódicas y asegurar el proceso de forma controlada, limitándolo a unos periodos concretos de actualización*

Podemos ver la lista de los repositorios configurados en un vitalinux si ejectuamos **Software y actualizaciones**. Repositorios Vitalinux Podemos observar los repositorios comentados. *NO es recomendable eliminar dichos repositorios para preservar la integridad del sistema. De hecho en Vitalinux se recrean en cada actualización por seguridad*. También podemod añadir un volumen (repositorio local en base a un CD, en lugar de la nube) o un repositorio (Añadir...), pero recomendamos lo siguiente.

Como añadir nuevos repositorios

Muchas veces nos puede interesar añadir otras fuentes de software ya que en las que tenemos no está el software requerido o está en una versión que no es la que necesitamos. Es muy importante tener en cuenta que:

1. Se deben usar siempre un **repositorio de confianza**, para lo cual haremos referencia al propio desarrollador del software. El uqe el software sea libre, garantiza no tener que andar buscando software pirata que puede contener código malicioso
2. También puedes consultar información al grupo de soporte del programa vitalnux (***soportevitalinux@educa.aragon.es***) y ellos te añadirán el repositorio en tus equipos de forma automática, sin necesidad de hacerlo a mano (¡y en todos los equipos!). Para eso contamos con **migasfree**. Y si además es interesante para todos, se pondrá a disposición de toda la comunidad educativa.

Para añadirlo puedes usar el botón de añadir que viste en la imagen anterior. Se te pedirá la url donde se encuentra y listo, pero lo menjor es usar el comando **apt-add-repository** que nos va a permitir añadir repositorios, y si el mismo viene con una firma, se añadirá iguamente. Veamos un ejemplo. Queremos instalar obs-studio, pero no está en los repositorios (mentira, si que está, y no tendrías que hacer nada, pero es una prueba para verlo). Nos iríamos a la página de de obs oficial, y en su apartado de downloads para Linux, nos encontramos con unas instrucciones a ejecutar en una terminal (puedes abrir una en cualquier linux con la combinación CTRL+ALT+t)

```
sudo add-apt-repository ppa:obsproject/obs-studio
sudo apt update
sudo apt install obs-studio
```

Nos interesan las dos primeras instrucciones

1. Con la primera estamos indicando al sistema que **añada un nuevo repositorio**, correspondiente a obs-studio en ppa. PPA es un almacen que nos permite distribuir software personal de una forma sencilla. Admite integración contínua (cuando se libera nuevo código se generan nuevas versiones de los paquetes de software) y está muy extendido su uso en la comunidad de Ubuntu y Debian
2. Con la segunda le estamos diciendo al sistema que **actualice el listado de software disponible**...es decir, que vaya a todos los respositorios/almacenes y pregunte qué paquetes (software) hay disponbible, para luego poder instalarlos. Si el sistema no tiene un índice de paquetes disponibles, cuando vayamos a instalar un paquete no sabrá si existe ni donde está

La última linea procederá a realizar la instalación del software (y todos los paquetes/librerias que necesite). Pero eso lo veremos en breve.

Parte 3. Aplicaciones Libres

Cómo instalar software y ver sus fuentes

En éste apartado vamos a aprender a instalar software libre y hablaremos al final un poco sobre el código fuente.

Instalación de Software

Una vez que sabemos qué aplicación queremos usar/probar, vamos a instalarla. Para ello tendremos varios métodos

1. Centro de Software
2. Synaptic
3. Mediante migasfree
4. Manualmente mediante Vitalinux Play o un paquete deb. Forma gráfica
5. Línea de comandos. Instalación de aplicaciones de un repo o directamente descagando el deb

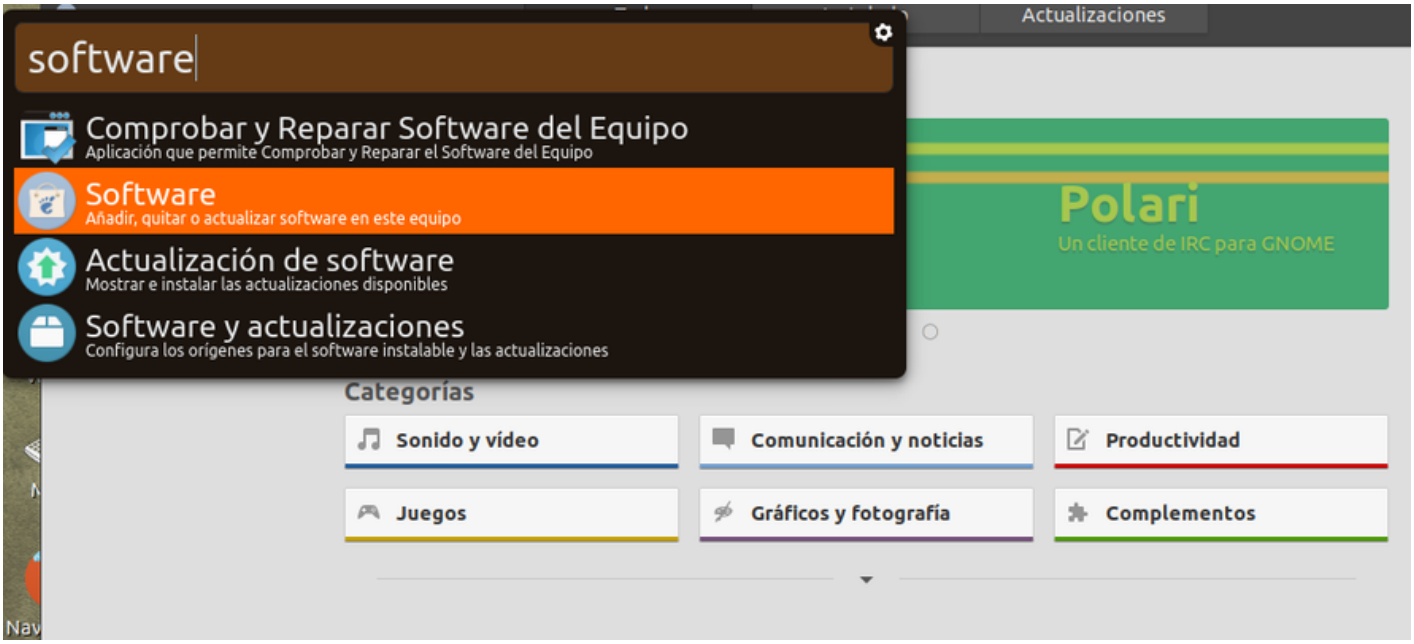
En la mayoría de los casos, deberemos hacer uso de una cuenta con permisos de administración para poder instalar/desinstalar software. En el caso de Vitalinux por ejemplo, la cuenta de profesor y dga tienen por defecto permisos de administración.

1. Centro de Software

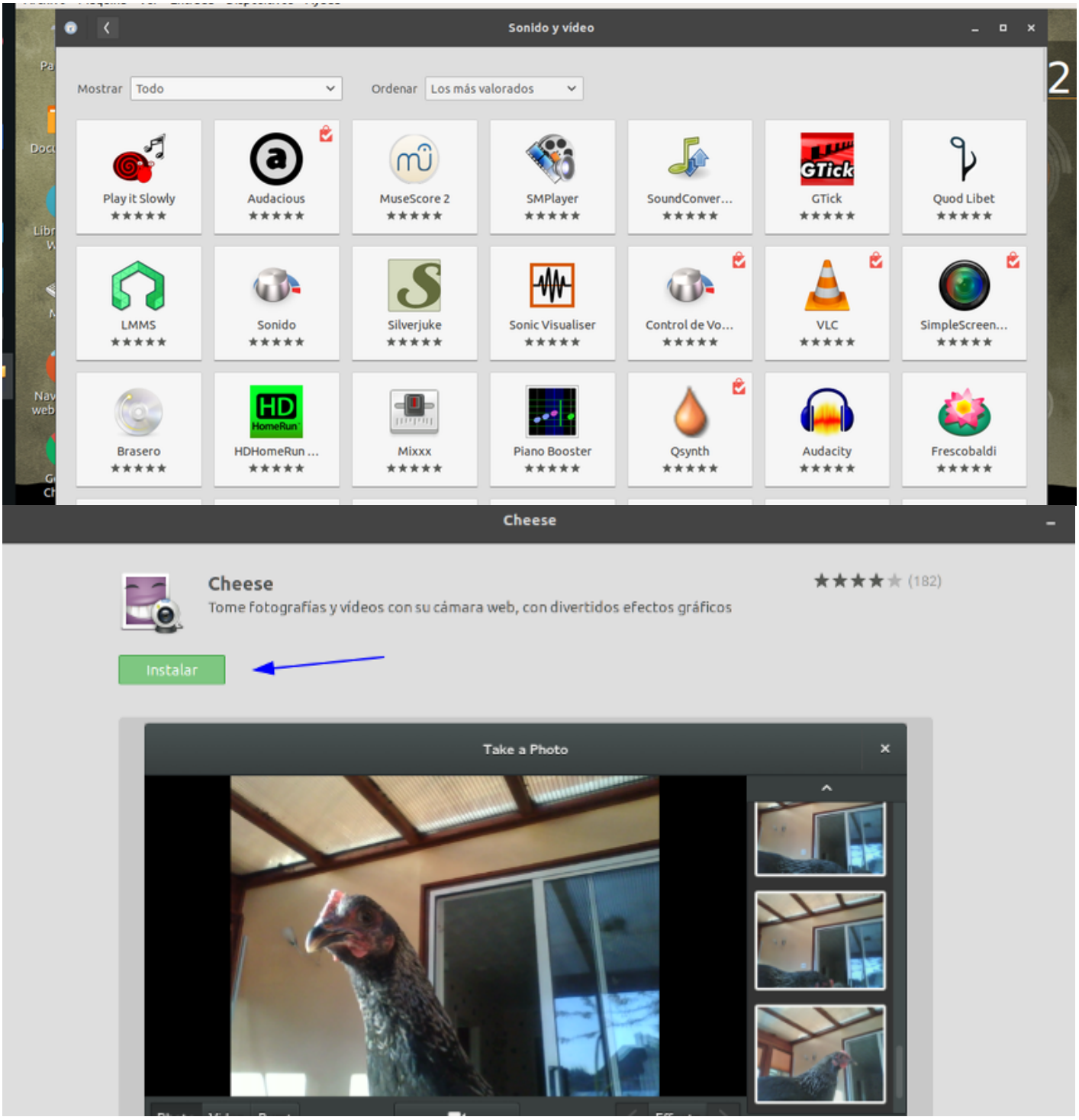
Vitalinux es una distribución basada en la versión ligera de Ubuntu, la cual dispone de su propio '''Centro de Software''' que permite al usuario instalar y desinstalar aplicaciones de una forma muy gráfica y convencional, al estilo que estamos acostumbrados de un Store actual del móvil, pudiendo filtrar la búsqueda de una manera categorizada: '''Educación, Ofimática, Internet''', etc.

> Atención: es posible que NO encontremos todo el software disponible (por problemas de indexación)

Para interactuar con '''El Centro de Software''' simplemente debemos lanzarlo pulsando la combinación mágia **CONTROL + ESPACIO** y tecleando **Software**. Si no disponemos del lanzador, lo podremos encontrar en el menú o con el nombre de Centro de Software.



Podemos entonces seleccionar la categoría, aplicación e instalar o incluso lanzarla (si ya está instalada)



2. Synaptic

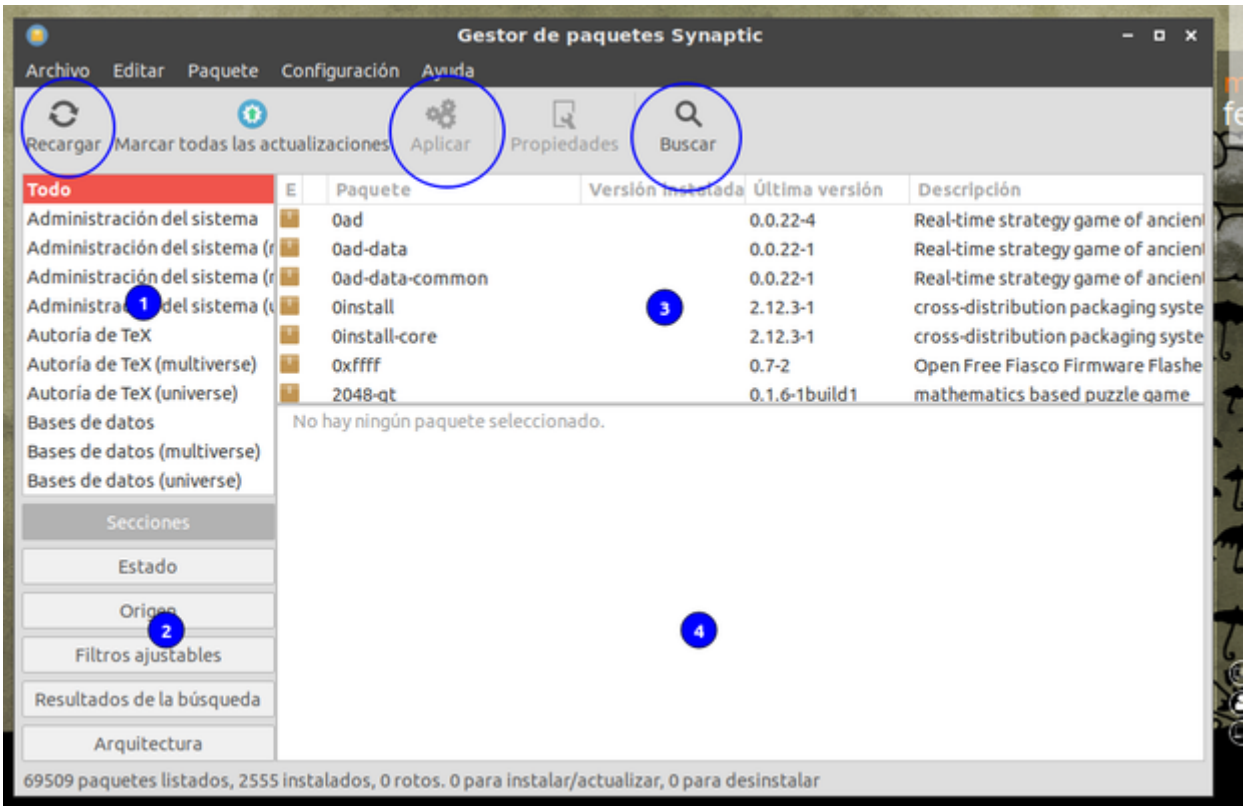
Una de las aplicaciones más completa (gráfica) y usada en el mundo Linux para la gestión del software es **Synaptic**. Aunque su aspecto visual no es tan amigable como el de otras aplicaciones equivalentes que han ido surgiendo con los años, su eficiencia y robustez han hecho que sea una

aplicación muy asentada y universalmente utilizada en muchas distribuciones Linux.

Entre sus características cabría destacar:

- Permite **actualizar la lista de software disponible en los repositorios** configurados en la máquina
- Permite **buscar** rápidamente, **instalar y desinstalar** aplicaciones
- Permite solventar problemas con **paquetes rotos** (paquetes que no se terminaron de instalar correctamente)

Para interactuar con **Synaptic** simplemente debemos lanzarlo pulsando la combinación **"CONTROL + ESPACIO"**' y tecleando **"Synaptic ..."**':



La pantalla de Synaptic está dividida en 4 secciones. Las dos más importantes son la lista que incluye la sección de categorías (1) en el lado izquierdo y la de los paquetes (3) en el lado derecho. Al seleccionar un paquete de la lista se mostrará una descripción del mismo (4).

Para **instalar** un paquete seleccionaremos una categoría, pulsaremos con el botón derecho del ratón sobre el paquete deseado y seleccionaremos **Marcar para instalar o bien haremos doble clic** en el nombre del paquete. Marcaremos de esta forma todos los paquetes que deseesemos instalar en el sistema y pulsaremos sobre el botón **Aplicar** para que comience su instalación. Synaptic descargará él solo los paquetes necesarios desde los repositorios

También es posible emplear el botón **Buscar** para encontrar los paquetes que deseemos instalar. Al pulsar sobre este botón podremos realizar búsquedas de programas mediante su nombre o descripción. Una vez localizado el programa que deseamos instalar hacemos doble clic sobre él para instalarlo. Si queremos eliminar un programa bastará con que pulsemos con el botón derecho encima de éste y **seleccionemos Eliminar o Eliminar por completo**.

En todos los casos, los cambios surtirán efecto una vez que hagamos clic sobre el botón de **Aplicar** .

3. Mediante migasfree

Como ya se ha dicho en varias ocasiones, una caracterísitica que vamos a encontrar en **Vitalinux** y que le diferencia del resto de distribuciones Linux actuales, es que incorpora un **cliente Migasfree** gestionado de forma centralizada y personlaziado por cada centro educativo, a demanda y de formma dinámica. En concreto, este cliente Migasfree garantiza que cada vez que arranca Vitalinux EDU DGA y tras iniciar sesión, se establece una comunicación con el servidor Migasfree del programa de **Software Libre** de la DGA, y a través de dicha comunicación Vitalinux deja en manos de **Migasfree** la configuración del equipo en cuanto a personalización y software: Migasfree provocará en el equipo Vitalinux todo aquello que se le haya encomendado previamente por el propio centro, y en concreto:

- Qué aplicaciones deben instalarse (y si no lo están, lo hará de forma automática)
- Qué aplicaciones se tienen que desinstalar (si las encuentra instaladas).
- Actualizar las aplicaciones instaladas y del sistema
- Instalación de periféricos: Impresoras, PDIs, Webcams....
- Además realiza otras acciones como:
- Configurar el entorno del escritorio
- Configurar las aplicaciones: navegadores, libreoffice, arduino...
- Limpieza del equipo
- Gestión de usuarios

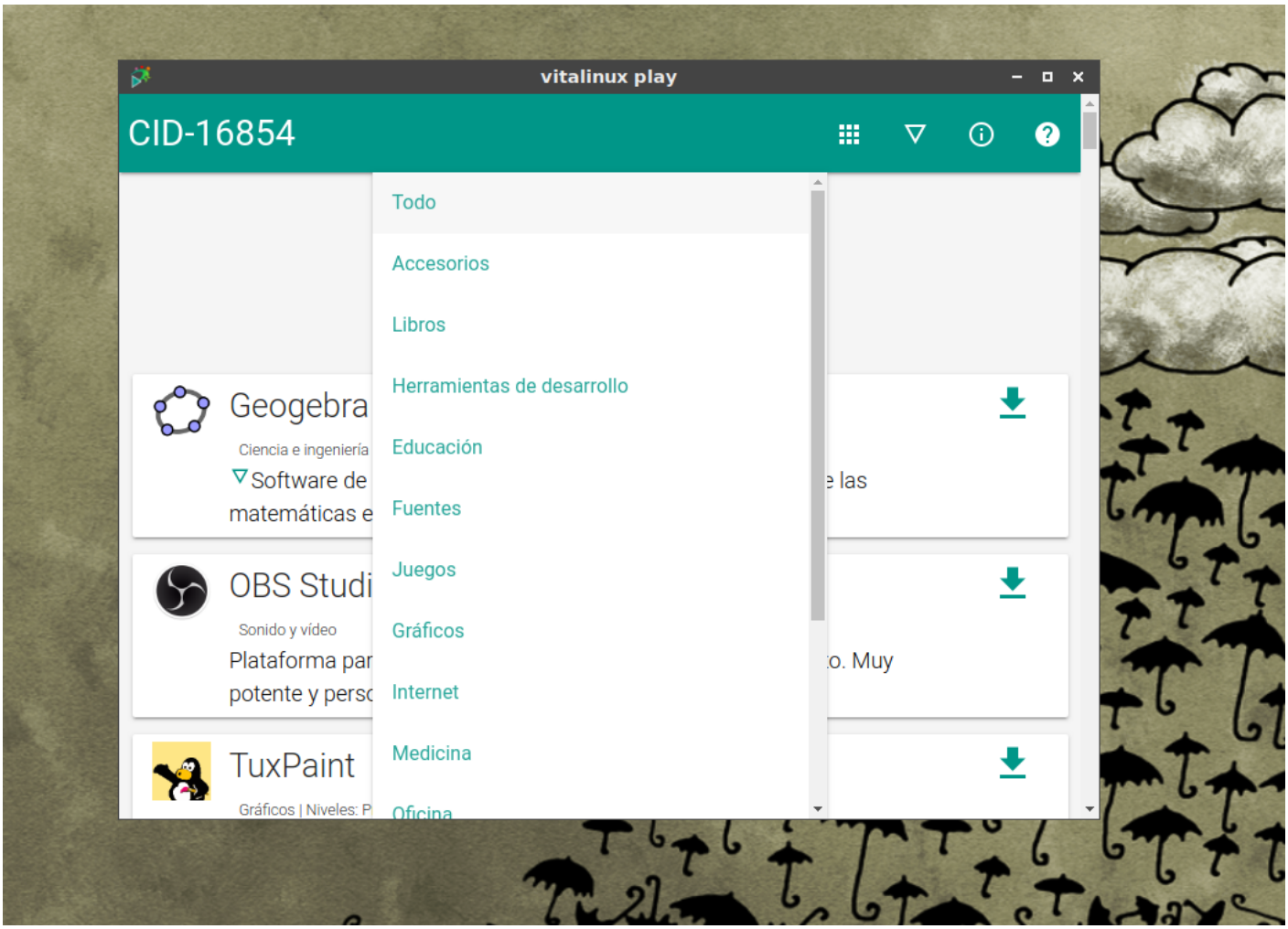
En definitiva, **cualquier aspecto que se pueda configurar vía software, se puede programar y automatizar para todos (o parte) de los eqiupos del centro**

¿Y cómo sabe qué tiene que hacer en cada equipo, si son muchos centros y cada uno usa un software diferente? A grandes rasgos, **Migasfree** tiene identificado al equipo Vitalinux en base a un identificador unívoco llamado **CID** y a un conjunto de **Etiquetas Migasfree** que se le pueden asignar. Cada centro puede personalizar así sus sistemas, y recibe la asesoría correspondiente entrando en el **programa de Asesoramiento del Proyecto Vitalinux**.

4. Forma manual

La forma más sencilla actualmente mediante la que cualquier usuario puede instalar software es usando **Vitalinux Play,** un catálogo de Aplicaciones Educativas o Generales que pueden resultar de interés para los docentes y estudiantes.

El catálogo es una selección que hemos elaborado y seguimos actualizando con las propuestas de todos los docentes y selecciona de todas las aplicaciones disponibles aquellas que pueden resultar de mayor interés, facilitando la búsqueda e instalación.



> Si queremos instalar un programa y no se encuentra en Vitalinux Play, la mejor opción es **abrir un ticket en soporte** a través del correo electrónico soportevitalinux@educa.aragon.es tal y como se indica en la propia aplicación, y desde allí nos contestarán y lo incluirán si es posible, lo que beneficia a toda la comunidad.

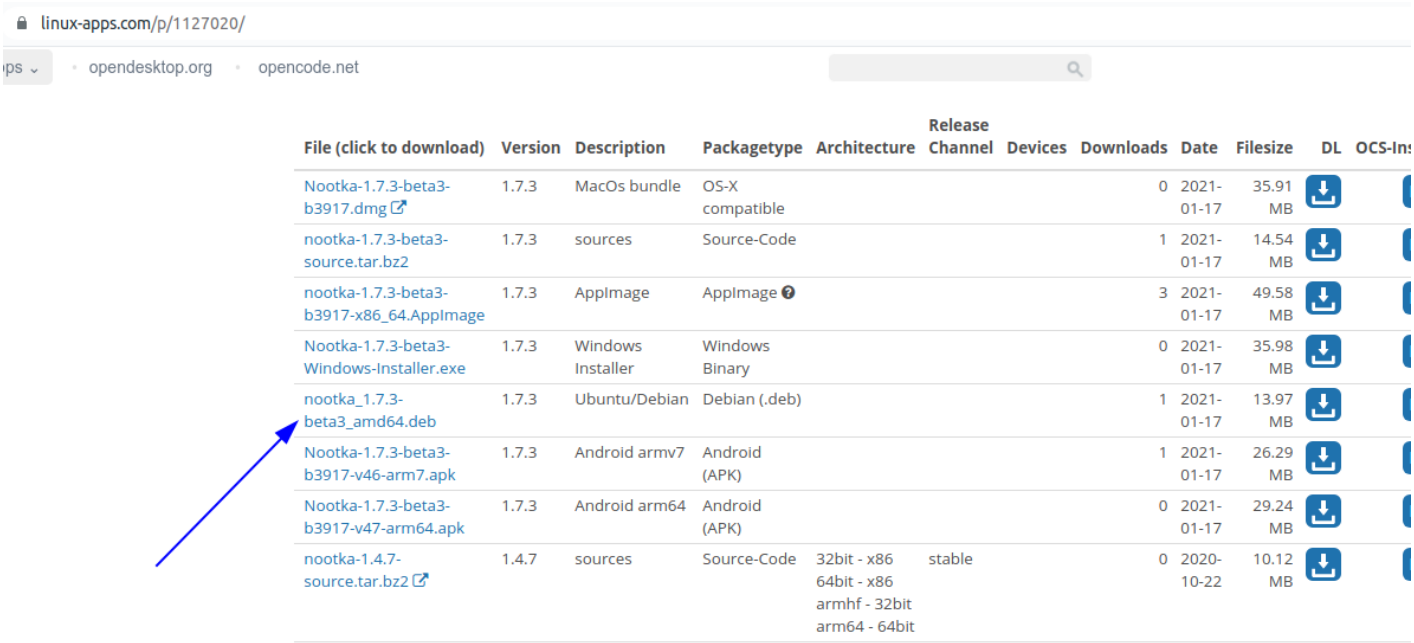
Si en cualquier caso, seguimos queriendo instalarlo de forma manual, dependiendo de cómo nos faciliten el software habrá que proceder de una forma u otra:

- O bien es un paquete que debemos "instalar" antes de poder usar. Como lo que hemos visto hasta ahora, pero de forma manual. Serán archivos **.deb**
- Es una aplicación que tiene todo los binarios y librerías ya compilados y empaquetados, de forma que se puede lanzar directamente. En éste caso, solo debemos marcarlos para ejectuar y lanzarlo. Es lo que se conoce comunmente como software portable. El más común actualmente es en formato **AppImage**

> Si optamos por la fórmula manual, debemos ser cuidadosos y saber de dónde estamos descargando el software para garantizar la seguridad e integridad de nuestro equipo y sobre todo, respetar las licencias de uso para respetar la legalidad del uso de la misma.

Veamos un ejemplo de cada:

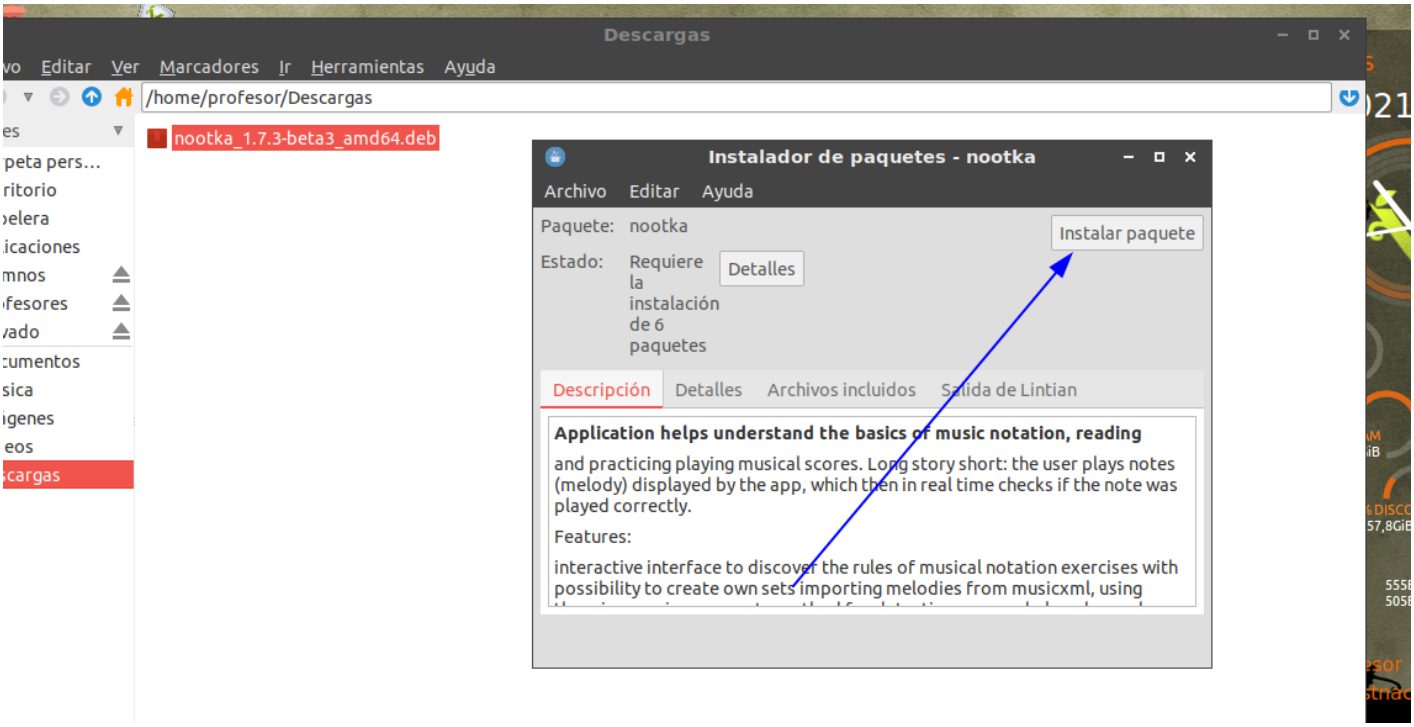
4.1 Paquetes deb



File (click to download)	Version	Description	Packagetype	Architecture	Release Channel	Devices	Downloads	Date	Filesize	DL	OCS-In
Nootka-1.7.3-beta3-b3917.dmg	1.7.3	MacOs bundle	OS-X compatible				0	2021-01-17	35.91 MB		
nootka-1.7.3-beta3-source.tar.bz2	1.7.3	sources	Source-Code				1	2021-01-17	14.54 MB		
nootka-1.7.3-beta3-b3917-x86_64.AppImage	1.7.3	AppImage	AppImage				3	2021-01-17	49.58 MB		
Nootka-1.7.3-beta3-Windows-Installer.exe	1.7.3	Windows Installer	Windows Binary				0	2021-01-17	35.98 MB		
nootka_1.7.3-beta3_amd64.deb	1.7.3	Ubuntu/Debian	Debian (.deb)				1	2021-01-17	13.97 MB		
Nootka-1.7.3-beta3-b3917-v46-arm7.apk	1.7.3	Android armv7	Android (APK)				1	2021-01-17	26.29 MB		
Nootka-1.7.3-beta3-b3917-v47-arm64.apk	1.7.3	Android arm64	Android (APK)				0	2021-01-17	29.24 MB		
nootka-1.4.7-source.tar.bz2	1.4.7	sources	Source-Code	32bit - x86 64bit - x86 armhf - 32bit arm64 - 64bit	stable		0	2020-10-22	10.12 MB		

1. Buscamos el software a instalar. Encontramos el fichero a descargar. El nombre del archivo (nootka_1.7.3-beta3_amd64.deb) nos está indicando la versión (1.7.3) y en éste caso la arquitectura (64 bits). Deberemos descargar éste si nuestra arquitectura del sistema operativo es 64 bits (lo mas común a día de hoy) o buscar el de 32 (en éste caso no hay posibilidad). También podemos encontrar el final con un all en lugar de amd64, lo que indica que sirve para cualquier arquitectura

> Observar que el software tiene versiones para múltiples arquitecturas...como debe ser :-)



1. Una vez descargado, podemos instalarlo simplemente haciendo doble click sobre el paquete. Al instalarlo nos puede saltar una advertencia indicando que dicho paquete ya está en un "canal de distribución" (como es el caso), es decir, que se encuentra disponible en los repositorios...por si queremos instalarlo de ésta forma y asegurar que viene firmado y estable a la versión de distribución que tenemos (por ejemplo, las posibles librerías que necesite estarán en la versión esperada).

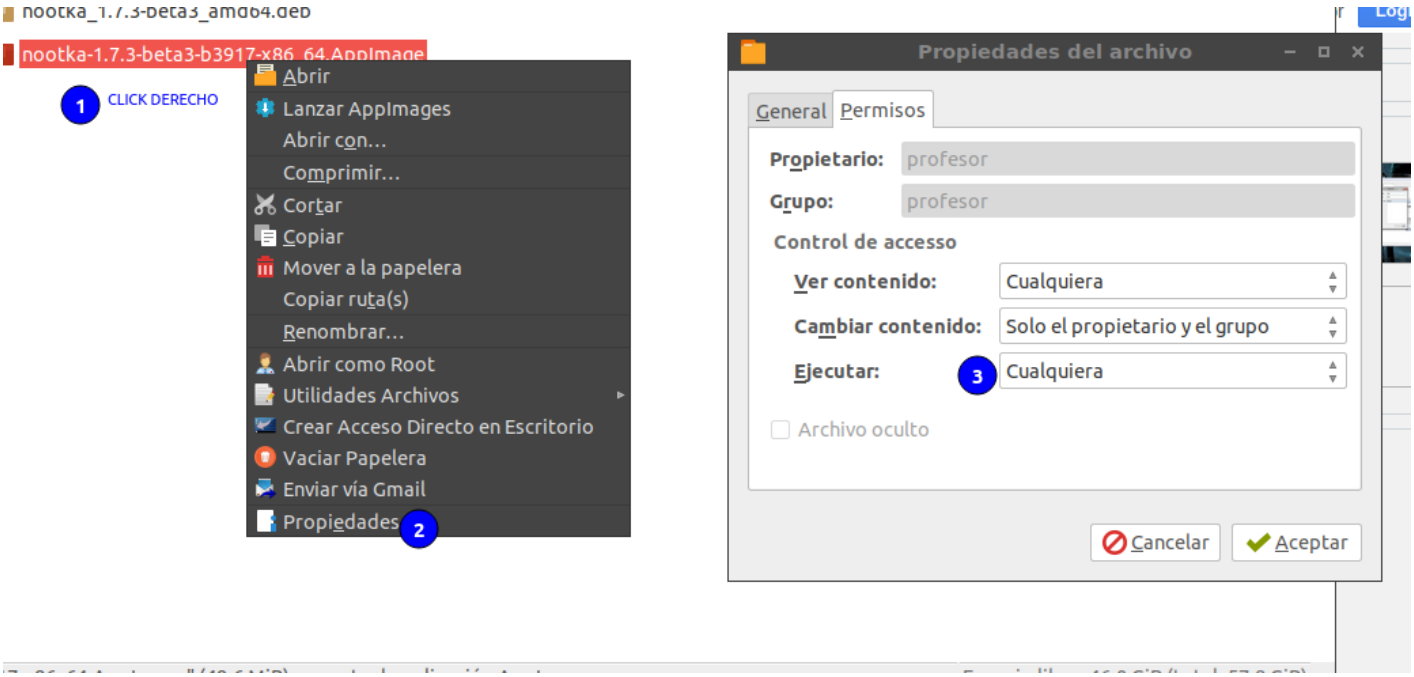
Es posible también encontrar url's en páginas web que directamente apuntan al paquete deb mediante un protocolo para instalar, siempre que sigamos el link desde un Linux. Lo podremos probar en la siguiente parte.

4.2 Ficheros AppImage

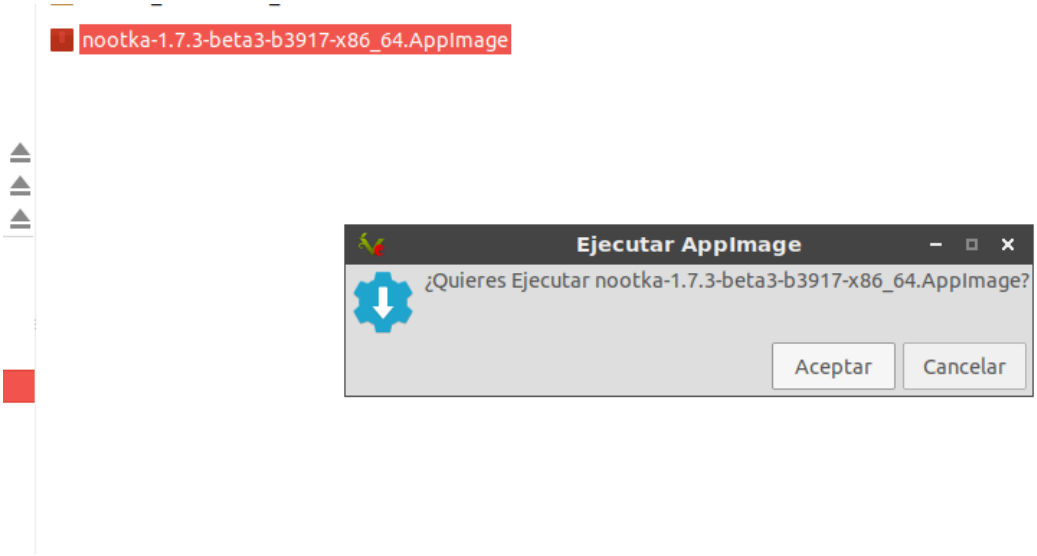
En el caso de ficheros AppImage, procederemos a descargarlo igualmente. En este ejemplo, el nombre sería nootka-1.7.3-beta3-b3917-x86_64.AppImage, indicando que es para una arquitectura de 64 bits

Ésta forma de distribución, no requiere de ninguna instalación para ejecutarse, pero:

- Es muy **cómodo** ya que no requiere instalación, dependencias de librerías con versiones que no tenemos
- No requiere permisos **privilegiados** para poder usarlo (dependiendo de las acciones que quiera realizar). Los paquetes deb requieren permisos de superusuario para ser instalados.
- **No** se mantendrá **actualizado**
- Es posible que le cueste arrancar algo más, ya que contiene todas las librerías y programas adicionales que necesite
- Deberemos añadir el **lanzador** que queramos para poder tenerlo accesible de forma cómoda (menú, escritorio, lanzador de aplicaciones...)



Es importante que demos **permisos de ejecución** al fichero descargado para poder ejecutarlo. Una vez hecho ésto, podremos lanzarlo con doble-click sobre el archivo descargado



> En vitalinux se ha facilitado su ejecución, de forma que no es necesario dar permisos de ejecución

Hay otras tecnologías de distribución de software como pueden ser Snap y Flatpak, que se están abriendo camino. Puedes consultar más información aquí por ejemplo sobre snap aquí

5. Línea de comandos

Mediante la línea de comandos, podemos instalar software, desinstalar, comprobar, buscar...usando los repositorios que tenemos configurados. Antes ya hemos visto algo con obs-studio. Veamos un ejemplo mas completo de acciones que puedes hacer

Actualizar la lista de paquetes disponibles en el repositorio

```
sudo apt-get update
```

Actualizar todos los paquetes instalados en el equipo

```
sudo apt-get upgrade
```

Instalar un paquetes:

```
sudo apt-get install nombre_del_paquete
```

Desinstalar paquetes:

```
sudo apt-get remove nombre-del-paquete
```

Desinstalar un paquete y sus archivos de configuración asociados:

```
sudo apt-get remove --purge nombre-del-paquete
```

Código fuente

Para finalizar éste apartado, debemos destacar un aspecto importante del software libre: es **código abierto**, y por tanto podemos consultar su código fuente, modificarlo...

> Recordar que OpenSource no siempre es software libre. Podemos tener programas de código abierto (opensource) con licencias restrictivas de uso.

Podemos descargar el código fuente usando el siguiente comando, siempre que **tengamos en los repos** marcado que queremos tener disponible además del software (deb) el código fuente (**deb-src**)

```
apt source nombre_paquete
```

Pero una forma más interesante para poder consultar el código es ir a las plataformas que se usan para la gestión de proyectos de software. La mayoría usa un sistema de control de versiones llamado git. Como servidores, hay muchos, pero los más usadas son:

- Launchpad. Por ejemplo del paquete Cheese
- Github. Aquí encontramos por ejemplo el código del software comentado antes obs-studio
- Gitlab. Por ejemplo el código de Inkspcape

Está fuera del alcance de éste curso manejar éstas herramientas de control de código. Conocer de algunos proyectos/aplicaciones el código fuente, o por lo menos seguir su desarrollo nos va a permitir involucrarnos mas en dicho programa, estar atento a las novedades, poder reportar incidencias o incluso colaborar en su mejora. Dependiendo de la edad de los alumnos y su compromiso puede ser interesante incluso hacerles participes a los mismos. Por ejemplo...podríamos ayudar a traducir un software al castellano como han hecho algunos COFO's en sus centros...estaríamos fomentando las habilidades lingüísticas y tecnológicas.

El **código fuente de los programas que se escriben para Vitalinux**, lo podemos encontrar en su propio servidor de gitlab

Para terminar, puedes visualizar un vídeo sobre la gestión del Software en Vitalinux, donde se habla del uso de las diferentes herramientas descritas

https://www.youtube.com/embed/8tBh8yz1FHY

Fuentes: * https://itsfoss.com/remove-install-software-ubuntu/ * https://ubunlog.com/

Parte 3. Aplicaciones Libres

Aplicaciones comunes. Breve Guía

En ésta última parte vamos a descubrir algunas aplicaciones libres. En Vitalinux podemos disponer de todas las aplicaciones y utilidades que encontraríamos en cualquier otro Ubuntu, y muchas otras aplicaciones que no están disponibles por defecto.

El ecosistema de aplicaciones disponibles puede llegar a ser abrumador. Descubre en los siguiente enlaces algunos ejemplos:

- Linux Apps específicas para educación ordenadas por fecha de publicación
- Blog desde Linux
- Alternativeto

Recuerda que puedes encontrar estas aplicaciones y otras muchas a través de la herramienta de gestión de paquetes Synaptic y mucho más adaptadas al mundo educativo a través de **Vitalinux Play**.

Veamos algunos ejemplos de aplicaciones **disponibles**, por **categorías** y para finalizar veremos **ejemplos de aplicaciones** que se usan en Aragón con la experienca de los propios usuarios, ya que nos puede dar una aproximación mejor de dichas aplicaciones, o indicaciones de las que usan en otras comunidades autónomas.

Copyright 2026



Alpha Stock Images - http://alphastockimages.com/. Creative Commons 3 - CC BY-SA 3.0

1. Aplicaciones disponibles

Como Vitalinux se basa en Ubuntu, las **aplicaciones disponibles** para el sistema serán las que aparezcan en dicha rama. Pero desde el equipo de Vitalinux se ha configurado el sistema para que se **actualicen** algunas de éstas con versiones más modernas y tener disponibles **otras aplicaciones** que no están disponibles por defecto en Ubuntu:

- **Libreoffice** (v.6.4): Suite ofimática para procesamiento de textos, hojas de cálculo, presentaciones o edición de documentos por imágenes como publisher. Se puede poner la visualización tipo ribbon (por pestañas).
- **Navegadores** (Firefox, Chromium, Chrome, Brave, Midori...) y complementos Java actualizados.
- Flash ha dejado de dar soporte, pero en Vitalinux se ha creado una herramienta para **lanzar recursos flash** mientras se sigan necesitando
- Aplicaciones de almacenamiento en la nube
- **Drivers para impresoras** en red y local de casi todas las marcas y modelos: HP, Epson, Canon... También con códigos
- Drivers y aplicaciones de las **PDI's** de forma cómoda para el centro

- **Aplicaciones educativas** o de terceros que no está paquetizadas por defecto para Ubuntu, pero que ahora lo están de forma sencilla...

Importante: En **Vitalinux** puedes encontrar aplicaciones que empiezan por vx-dga-l-xxxx. Éstas son o bien aplicaciones que ha desarrollado el equipo de soporte, o bien aplicaciones de terceros que no venían disponibles para la versión de Ubuntu de base y que se han "repaquetizado". Hay varios ejemplos: musescore, qcad, arasuite, aplicaciones de editoriales...

2. Aplicaciones por categorías

Veamos algunos ejemplos por **categorías**. Luego veremos ejemplos de aplicaciones que se usan en Aragón con la experienca de los propios usuarios, ya que nos puede dar una aproximación mejor de dichas aplicaciones, o indicaciones de las que usan en otras comunidades autónomas.

Puedes clickar en el vínculo de **instalar** y te instalará la aplicación, si tienes la suerte de estar en un Linux/Vitalinux :-)

Imagen

- GIMP. El programa de procesado de imágenes libre más usado. Instalar
- Darktable. Laboratorio fotográfico digital. Instalar
- Krita. Parecido a GIMP pero más orientado al dibujo. Instalar
- Inkscape. Dibujo vectorial libre. Sustituto de Illustrator, FreeHand, etc. Instalar
- Sweet Home-3D. Diseño de interiores 2D y 3D. Instalar
- SIR (Simple Image Resizer). Para cambiar el tamaño y otros parámetros de una imagen; admite procesado por lotes. Instalar
- Entangle. Para hacer fotografías con una cámara controlada desde el ordenador por USB, y previendo la imagen en la pantalla del ordenador. Instalar
- RawTherape RawTherape es un sistema de revelado digital para procesar imágenes RAW.

Vídeo y multimedia

- Openshot. Editor de videos en su versión más reciente. Multiplataforma (Linux, Windows y Macintosh). Instalar
- VLC. Reproductor de videos. Instalar
- Shotcut. Editor de video con algo más de potencia y posibilidades que Openshot. Multiplataforma. Instalar
- Open Broadcaster Sotware. Creación de vídeos y streaming. Instalar
- LightWorks. Estación de trabajo multimedia. Completa y compleja herramienta de produción multimedia. Nada que envidia a otras grandes. No es software libre, pero tiene una versión libre...nada que envidiar a las grandes Instalar
- Handbrake. Para ripear DVDs y convertir entre formatos de vídeo. Instalar
- Sozi. Presentaciones tipo Prezi, ver instrucciones. Instalar
- Selene. Conversor de audio y video. Instalar
- Kodi. Centro multimedia. Multiplataforma. Instalar

Sonido

- Musescore. Editor de partituras. Instalar
- Audacity. Editor de sonido más usado en educación. Instalar
- LMMS. Let's may music, vamos a hacer música. Para componer música mezclando MIDI y audio. Multiplataforma. Instalar
- Ardour. Una estación de trabajo para trabajos de Audio. Grabación, Edición y Mezcla. Muy potente. Instalar
- Mixxx. Mezclador de música para DJs, para amenizar las fiestas del cole/insti. Es multiplataforma. Instalar
- Lilypond. Lilypond es un programa de edición de partituras que sigue la tradición del grabado clásico de música Instalar
- TuxGuitar TuxGuitar es un editor y reproductor de tablaturas de guitarra multipista. Alternativa libre a herramientas como Guitar Pro. Instalar

Astronomía

- Stellarium: para enseñar lo básico sobre los planetas y constelaciones en cualquier materia (mitología, por ejemplo) y ver cómo estará el cielo esta noche o el día y la hora en que nacimos. Multiplataforma. Instalar
- Celestia: Para ver explorar el universo en casa. Multiplataforma. Instalar

Robótica y electrónica

La Robótica puede requerir de configuraciones adicionales. Se recomienda visitar ántes la página de la wiki.

- Mblock. Para programar el robot mbot. Instalar
- Web2board. Aplicación para poder programar con Bitbloq (placas por ejemplo BQ y otras muchas...). Instalar
- Bitbloq ''offline''. Por defecto se usa la aplicación en modo ''online'' (a través de navegador, pero se puede usar de forma ''offline''). Instalar
- Fritzing. Programar para Arduino. Instalar
- Arduino. Instalar
- Scratch y Scratch 2 ''offline''. Presinstalado por defecto
- S4A. Una modificación de Scratch. Instalar
- Logisim, simulador de circuitos electrónicos digitales. Instalar

Programación

- Thonny. IDE de programación para Pyhton sencillo. Instalar
- Brackets. Editor de código para JavaScript, HTML, CSS. Instalar
- Geany. Editor de texto ligero para programación; usa bibliotecas GTK. Multiplataforma. Instalar
- Gdevelop. Para crear juegos sin saber programar. Instalar
- Netbeans. Programación en Java. Instalar
- Appinventor. Instalar
- Notepadqq Editor de texto avanzado y rápido. Muy productivo para programación. Instalar

Copyright 2026

Libros digitales

Los libros digitales suelen llevar licencias más cerradas de uso, pero desde Vitlainux se facilita su instalación (con permiso de la editorial) para no tener que instalar los libros o aplicaciones a mano...puedes ver mas información aquí.

Otras aplicaciones

- Arasuite y Araword. AraSuite es un conjunto de herramientas para trabajar con personas con discapacidades, especialemnte si estas tienen relación con la comunicación. Requiere el uso de sistemas de comunicación aumentativos que combinan pixtogramas y palabras como conceptos relacionados. Instalar
- gElemental. Tabla periódica de los elementos. Instalar
- GeoGebra: Imprescindible en matemáticas. Instalar
- Calibre. Biblioteca de ePubs ordenada, con conversión entre formatos. Instalar
- Cura. Impresión en 3D avanzada. Instalar
- SimpleScreenRecorder. Grabación de videotutoriales. Instalar
- TBO. Crea tus propios tebeos. Instalar
- Xmind. Para crear mapas conceptuales. Instalar
- Processing. Instalar
- Cmaptools. Instalar
- Scribus. Edición y maquetación de escritorio. Instalar

Coordinación TIC

- Remmina. Para abrir sesión de escritorio remoto en otro ordenador de la misma red. Instalar
- Anydesk. Para control remoto (soporte), pero que también se puede usar para controlar equipos del centro o incluso mostrar en pantalla dispositivos móviles. Viene preinstalado en Vitalinux

Copyright 2026

- Epoptes. Programa de supervisión y control de los ordenadores de un aula conectada en red. Servidor - Instalar. Cliente - Instalar
- Filezilla. Cliente de FTP. Instalar
- Transmission. Programa de descarga de torrents. Multiplataforma. Instalar
- Virtualbox. Programa de virtualización de sistemas operativos. Se pueden descargar sistemas preinstalados de [https://www.osboxes.org/ OSBoxes.org] para bajar y probar. Instalar para VX-2.0
- WireShark. Para analizar el tráfico de red. Instalar

3. Experiencia con aplicaciones



Unsplash bank images

En la siguiente página de la wiki, podemos encontrar una tabla con Aplicaciones, clasificada por **Nivel Educativo** y **Posible Área de aplicación**, con una **descripción y experiencia** directa del propio profesor/coordinador.

Por otro lado, también podemos encontrar un listado categorizado de aplicacione educativas recomendadas en las Comunidad de Madrid para su distribución MAX.

4. Equivalencias con aplicaciones conocidas

Hoy día gracias a buscadores podemos encontrar alternativas al software privado es tan sencillo como realizar una correcta búsqueda (''español e inglés'') y perder algo de tiempo en probar lo que vamos encontrando. No obstante, a modo de ejemplo indicamos las siguientes páginas:

- Alternativeto: buscador de alternativas al software privativo. Además podemos ver si es software libre o no, o qué licencia tiene.
- Jueduland: muy buena, por categorías.
- Linux Adictos: alternativas de uso cotidiano; con capturas de pantalla y breve explicación de las aplicaciones.
- Desde Linux. Recopilación de aplicaciones del 2018/2019 de éste blog de noticias.
- Desde Linux Comparativa: Listado de comparativa con aplicaciones conocidas de otro sistema operativo.

Créditos

Autoría: Arturo Martín Romero e Ignacio Sancho Morte

Cualquier observación o detección de error en soporte.catedu.es

Los contenidos se distribuyen bajo licencia **Creative Commons** tipo **BY-NC-SA** excepto en los párrafos que se indique lo contrario.

Créditos

Curso creado en Junio de 2023 por:

Arturo Martín Romero e Ignacio Sancho Morte

Cualquier observación o detección de error en soporte.catedu.es

Los contenidos se distribuyen bajo licencia **Creative Commons** tipo **BY-NC-SA** excepto en los párrafos que se indique lo contrario.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU