

migas free

Fun with migasfree

Alberto Gacías

28 de julio de 2018

Índice general

1. Introducción	3
1.1. Acerca de este libro	3
1.2. Gestión de la Configuración Software	8
1.3. Administrando escritorios	11
1.4. Características de migasfree	18
2. Primeros pasos	21
2.1. Probando migasfree	21
2.2. Configurando software al estilo migasfree	28
2.3. Configurando migasfree-client	35
3. Guía de uso	41
3.1. La configuración del sistema migasfree	41
3.2. La Liberación	52
3.3. La sincronización de los sistemas	58
3.4. La Auditoría	60
3.5. Dispositivos	66
3.6. Ejemplos prácticos	70
4. Puesta en producción	75
4.1. Migasfree en producción	75
4.2. Creando tu propia Distro	78
4.3. FAQ	79
4.4. Resolución de problemas	84
5. Ajustes	87
5.1. Ajustes del servidor migasfree	87
5.2. Ajustes del cliente migasfree	93
6. Empaquetado	99
6.1. Empaquetando migasfree	99
6.2. Empaquetando proyectos python	101
7. Anexos	103
7.1. Bibliografía	103
7.2. Referencias	103
7.3. Glosario de términos	103

7.4.	API	104
7.5.	GNU Free Documentation License	125

A Patricia.

El deber de un ciudadano no es creer en ninguna profecía del futuro, sino actuar para realizar el mejor futuro posible.

—Richard Stallman.

La hipótesis del gran cerebro.

1.1 Acerca de este libro

1.1.1 Licencia y Copyright

Fun with migasfree

Copyright (C) 2013 Alberto Gacías and contributors. Permission is granted to copy, distribute and/or modify this document under the terms of the *GNU Free Documentation License*, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled «GNU Free Documentation License».

1.1.2 Presentación

ALBERTO GACIAS
Eyo presenta Eyo
-FUN- WITH MIGASFREE

Hola. Soy Alberto Gacías. Bienvenidos al primer capítulo de «Alberto Gacías presenta diversión con migasfree». Durante las próximas páginas, usted y yo vamos a explorar el dinámico mundo de la migasfreología.¹

¹ Recordando a Sheldon Cooper en «Fun with Flags» en la serie The Big Bang Theory.

¿Qué es migasfree en cuatro palabras?

Un gestor de despliegues.

Me atrevo incluso con sólo una palabra: **Desplegador.**

Migasfree se ocupa principalmente del proceso de la liberación de software y de la posterior auditoría de los cambios producidos en los equipos como consecuencia de esa liberación.

Es una de las herramientas que estamos utilizando con éxito en [AZLinux](#), el proyecto de migración a escritorio libre del [Ayuntamiento de Zaragoza](#).

Este software se ha hecho indispensable en nuestro día a día, y creo que es una buena solución para personalizar y administrar escritorios de forma eficaz.

¿Para qué sirve migasfree?

- Para administrar ordenadores de forma centralizada.
- Desplegar actualizaciones según calendarios y características de ordenadores y usuarios.
- Controlar y personalizar cada ordenador, manteniendo íntegros tus sistemas.
- Auditar hardware y software.

Este libro te introducirá en el uso de migasfree y lo escribo a medida que mejoramos el software, con lo que, si te lo descargaste hace tiempo, quizás ya esté obsoleto. Tenlo en cuenta.

1.1.3 A quién va dirigido

Este libro puede serte útil si eres administrador de escritorios (y/o servidores) y quieres personalizar y administrar de forma eficaz tus equipos, manteniendo la integridad de los sistemas.

1.1.4 Estructura

Introducción

En esta primera parte, repasaremos la Gestión de la Configuración del Software. Conocer los aspectos básicos de este proceso de la Ingeniería del Software te dará una visión de conjunto que considero esencial porque es, precisamente aquí, donde se integra migasfree.

Te explicaré las dificultades que un administrador de escritorios va a encontrarse y cómo se pueden sortear de forma sencilla, basándome en la experiencia adquirida en AZLinux.

Podrás conocer la historia, características y componentes que utiliza migasfree.

Primeros pasos

Aquí te enseñaré a instalar y probar un servidor y cliente migasfree, con la configuración mínima, para que puedas verlos en funcionamiento cuanto antes.

Guía de uso

Te permitirá conocer tanto el cliente como el servidor migasfree más en detalle.

Puesta en producción

Se tratarán los aspectos a tener en cuenta si quieres utilizar migasfree en un entorno de producción, así como las FAQs y la resolución de problemas.

Ajustes

Detalla los ajustes necesarios para configurar correctamente tanto el servidor migasfree como los clientes.

Empaquetado

Contiene intrucciones para empaquetar migasfree en cualquier distribución.

Anexos

Contiene la API de migasfree, la bibliografía, referencias y licencia de este libro.

1.1.5 Agradecimientos

Detrás de cada proyecto hay personas que lo hacen posible, manteniendo, animando, corrigiendo, colaborando, apoyando...

Deseo expresar en primer lugar mi gratitud a Eduardo Romero. Me dio el estímulo necesario para liberar la primera versión de migasfree, haciendo visible este proyecto en Internet. También aportó la primera y única donación que ha recibido migasfree (aunque fuera por una apuesta perdida, no se lo tuve en cuenta y fue muy bien recibida).

A Jose Antonio Chavarría, compañero de fatigas (y alegrías), también me siento agradecido. Ha sido y es piedra angular en migasfree. Ha mejorado sustancialmente el proyecto reescribiendo el código *spaguetti* a buen código², aportando ideas y soluciones. Me tranquiliza cuando quiero correr en exceso, y es el guardián de la simplicidad de migasfree.

A Jesús González por su empeño en crear equipos de trabajo donde las personas nos sentimos a gusto trabajando.

Y a todo el grupo de Asistencia a Usuarios del Ayuntamiento de Zaragoza, especialmente al equipo de Software Libre. Disfruto trabajando con ellos y me hacen reír a diario.



De izq. a dcha.: El Pequeñico, Grandullón, El Mimo, El Puto Amo, Dios, The Boss, El Hippie y El Burbujulas.

Figura 1: Grupo de Software Libre del Ayuntamiento de Zaragoza.

² Proceso conocido muy localmente como *chavarrización*.

1.1.6 Acerca de mí

De joven me atraía la programación. Estudié electrónica, y allí me enseñaron a programar en código máquina el microcontrolador 8751.

Aún me gusta cacharrear con transistores, condensadores, circuitos integrados, leds... y enseñar lo poco que recuerdo de todo aquello a Jesús. A los dos nos gusta jugar con [arduino scratch](#) y [s4a](#).

Trabajé como electrónico mis primeros años laborales y, poco después, con el boom de la informática personal, empecé a desarrollar aplicaciones de todo tipo.

Actualmente trabajo como técnico informático en el equipo de Software Libre del Ayuntamiento de Zaragoza desarrollando y manteniendo AZLinux, el escritorio libre que usamos los trabajadores municipales.

Parte de mi tiempo libre lo dedico a desarrollar [migafree](#).

Me encanta mirar el cielo en las noches de verano de Peñíscola, la cerveza, las migas y los huevos rotos.

También me gusta escuchar música, el olor a tierra mojada y que me hagan reír con cualquier tontada.

Amo a Patricia, y a Jesús, nuestro hijo.

1.1.7 Nota del autor

Algunos han criticado a las personas que entregamos parte de nuestro tiempo en producir software libre. El argumento se basa en que nuestra aportación hace que se eliminen puestos de trabajo o que algunas empresas no pueden hacer negocio por competencia desleal. Dicen estar hartos de gente que «trabajamos gratis» y que les «quitamos» el sustento.

No puedo estar de acuerdo. En primer lugar porque el [software libre](#) no es un asunto económico sino que, sencillamente, lo que plantea es una cuestión de **libertad**.

Pienso que los modelos de negocio obsoletos deben adaptarse y evolucionar hacia nuevas formas de generar riqueza, creando nuevas relaciones entre productor y consumidor. Los modelos de negocios basados en software libre a menudo nos indican el camino a seguir, ya que están estableciendo estas nuevas relaciones y obteniendo la confianza y el reconocimiento del consumidor, y no precisamente por cuestiones económicas. ¿No desean esto las empresas para sí?

En cuanto a que producir algo que otros obtienen gratis no genera puestos de trabajo, pienso que es falso. Sólo hace falta fijarse en como las tecnologías de interconexión, protocolos y servicios de accesibilidad de la red Internet ha generado, y seguirá generando, infinidad de puestos de trabajo. Estoy convencido que Internet no sería ni siquiera una sombra de lo que es si estas tecnologías se hubieran patentado, cerrado y/o explotado económicamente.

Creo que el movimiento del software libre es, junto con otros, una esperanza para que el Conocimiento vuelva a ser producido por la sociedad y para la sociedad, en contraposición al Conocimiento creado, comercializado y controlado por determinadas organizaciones y que, en ocasiones, causa un perjuicio a la sociedad.

1.1.8 Enlaces

- [Versión en HTML](#).
- [Versión en PDF](#).
- [Ejemplos usados en el libro](#).
- [Código fuente](#).
- [Proyecto migafree](#).
- [Twitter: @migafree, @albertogacias](#).

1.2 Gestión de la Configuración Software

Nada es permanente a excepción del cambio.

—Heráclito de Éfeso.

Estamos acostumbrados a actualizar periódicamente nuestras aplicaciones: los sistemas se hacen obsoletos rápidamente, aparecen nuevas tecnologías, hay errores que son resueltos, surgen nuevas necesidades. Sin importar en qué momento del ciclo de vida del sistema nos encontremos, el sistema cambiará, y el deseo de cambiarlo persistirá a lo largo de todo el ciclo de vida.³

Por tanto, el cambio en el software es **inevitable** y es **deseable** que ocurra.

Es inevitable porque los desarrolladores cometemos errores y es mediante una modificación como los corregimos. A este tipo de cambios los llamamos **correctivos**.

Por otro lado, el cambio es deseable ya que a menudo queremos incorporar nuevas funcionalidades al software o mejorar aquellas que ya existían. Mediante los cambios **evolutivos** es como mejoramos el software.

El cambio genera **confusión** e incertidumbre y se produce desde que concebimos, construimos y también mientras mantenemos un proyecto software.

El gran reto reside precisamente en gestionar de forma controlada dichos cambios usando alguna estrategia que los favorezca y facilite.

De esto trata precisamente la **Gestión de la Configuración Software** (GCS), un proceso de la Ingeniería del Software que identifica, hace seguimiento y controla cada uno de los cambios que se producen en los sistemas.

1.2.1 Objetivo

El objetivo de la GCS es **conservar la integridad** de los sistemas frente a los cambios.

Un sistema será íntegro frente al cambio si:

- Mantiene correctamente las **relaciones** entre los distintos cambios a medida que se van produciendo (el típico problema de dependencias entre elementos).
- Permite la **auditoría** de cambios (conocimiento del estado de un sistema al que se le han ido aplicando cambios sucesivamente).

1.2.2 El proceso

El proceso de la GCS es un conjunto de actividades que nos permitirá garantizar dicha integridad, y que podemos resumir en:

- Petición de cambio.
- Cambio.
- Liberación.

Petición de cambio

Cuando se nos reporta un error o una petición de mejora, lo primero que hacemos es identificar el **Elemento de Configuración Software** (ECS) al que se refiere.

³ Primera ley de la Ingeniería de Sistemas, Software Configuration Management, Bersoff, Henderson & Siegel, Prentice-Hall, 1980

Un ECS es cualquier objeto software sometido a la GCS. Puede ser un manual de usuario, una especificación, un conjunto de datos para realizar tests, una aplicación, una librería, incluso las herramientas que se usan para realizar dichos cambios, etc.

Una vez identificado el ECS se registra la petición de cambio.

Las herramientas típicas para registrar y hacer el seguimiento del cambio son los denominados *gestores de proyectos* (Redmine, Bugzilla, Tracker, etc.)

Cada petición de cambio es analizada más tarde pudiendo ser aceptada o rechazada. Si es rechazada, se avisa al informador y se cierra la petición. Si es aceptada, se asigna la petición a alguien para que realice dicho cambio.

Cambio

El cambio es la actividad que modifica el ECS, generando una nueva versión del ECS.

En esta actividad se utilizan un conjunto muy diverso de herramientas, desde procesadores y editores de texto, sistemas de control de versiones, entornos de desarrollo integrados (IDE), depuradores, compiladores. . .

Liberación

La liberación es la actividad de situar la nueva versión del ECS generada, en un repositorio o almacén para que posteriormente los clientes del ECS puedan acceder a él e instalarlo.

1.2.3 Elemento de Configuración Software

Si observamos como los diferentes proyectos de Software Libre realizan la GCS, vemos que realizan las actividades mencionadas liberando finalmente el código fuente del proyecto en Internet.

Estos proyectos trabajan con distintos tipos de ECS (.png, .txt, .py, .c, .bin, etc.) usando los **Sistemas de Control de Versiones**, junto con las plataformas de desarrollo colaborativo como sourceforge.net, github.com, etc.

Este código fuente será posteriormente compilado por los mantenedores de las distribuciones GNU/Linux (Fedora, Red Hat, Debian, Ubuntu, etc.), realizando su propia GCS. Pero a diferencia de los primeros, las Distribuciones GNU/Linux sólo trabajan sobre un único tipo de ECS: **el paquete**, donde introducirán el programa ya compilado.

Este simple hecho permite garantizar la integridad frente a los cambios de forma eficaz y sencilla, como veremos a continuación.

El paquete

Un paquete es un contenedor que encapsula un conjunto de ECS liberados por un determinado proyecto, junto a su metainformación.

Contendrá, por tanto, el programa compilado para una determinada distribución y arquitectura, más un amplio conjunto de información, como puede ser:

- El autor del programa.
- La dirección del repositorio del proyecto.
- La versión del ECS.
- La arquitectura.
- El nombre y dirección e-mail del empaquetador.
- La fecha de empaquetado.

- El nombre del equipo en que se produjo el empaquetado.
- Una descripción corta del contenido del paquete.
- Una descripción larga.

Pero, además, suelen incluir:

- Código a ejecutar antes y después de:
 - Instalar.
 - Actualizar.
 - Eliminar el paquete.
- Dependencias con otros paquetes.

Una vez que un mantenedor de una Distribución GNU/Linux ha creado un paquete, lo libera poniéndolo en un repositorio público a disposición de los clientes.

Puesta en producción

Los encargados de aplicar los cambios son los programas denominados **gestores de paquetes** tales como `yum`, `zypper` o `apt`.

Un gestor de paquetes es un programa que permite poner en producción los cambios que han sido liberados en los repositorios.

La actualización de un equipo se realiza comparando las versiones de los paquetes instalados con los paquetes de los repositorios públicos, detectando los que han aumentado su versión, resolviendo sus dependencias y finalmente, si no hay conflictos, obtienen desde los repositorios los paquetes necesarios.

Una vez han descargado los paquetes, dan órdenes a los **backends** (`rpm`, `dpkg`, etc.), para que se produzca la desinstalación de los paquetes antiguos y la instalación de los nuevos.

Los backends abren el paquete, y grosso modo:

1. Extraen los ficheros del programa copiándolos en el sistema, y ejecutan además el código programado para antes y después de la actualización.
2. La metainformación es extraída del paquete y se almacena en la base de datos del backend.

Decía Ian Murdock, fundador de Debian, que el gran aporte del software libre a la industria, ha sido precisamente la invención del sistema de paquetería (paquete, repositorio, gestor de paquetes).

Y no es para menos, ya que este sistema nos proporciona los dos requisitos necesarios que garantizan la integridad frente a los cambios:

1. El control de dependencias, mediante el gestor de paquetes.
2. La auditoría, mediante las consultas a la base de datos del backend.

Nota: Si estás acostumbrado a instalar programas mediante el típico “./configure, make, install”, tienes que ser consciente que estás rompiendo la integridad frente a los cambios, ya que la base de datos del backend no es actualizada con este procedimiento. Todo lo que no sea instalar programas mediante el gestor de paquetes o el backend, rompe la integridad.

A continuación, se muestra una tabla con las operaciones más comunes que puedes emplear sobre algunos de los gestores de paquetes más conocidos:

Operaciones Gestores	apt	yum	zypper
Actualizar listado de paquetes disponibles	apt-get update	yum check-update	zypper refresh
Actualizar sistema	apt-get upgrade	yum update	zypper update
Instalar paquete	apt-get install <pkg>	yum install <pkg>	zypper install <pkg>
Borrar paquete	apt-get remove <pkg>	yum erase <pkg>	zypper remove <pkg>
Buscar paquete	apt-cache search <pkg>	yum list <pkg>	zypper search <pkg>
Buscar repositorio de un paquete	apt-cache madison <pkg>	yum whatprovides <pkg>	zypper what-provides <pkg>

Veamos ahora las operaciones más comunes que puedes emplear sobre los backends `dpkg` y `rpm`:

Operaciones Backends	dpkg	rpm
Instalar fichero de paquete	<code>dpkg -i <file .deb></code>	<code>rpm -ivh <file .rpm></code>
Borrar paquete	<code>dpkg -r <pkg></code>	<code>rpm -e <pkg></code>
Paquete que instala el fichero <file>	<code>dpkg -S <file></code>	<code>rpm -qf <file></code>
Ficheros del paquete	<code>dpkg -L <pkg></code>	<code>rpm -ql <pkg></code>
Descripción del paquete	<code>dpkg -s <pkg></code>	<code>rpm -qi <pkg></code>
Paquetes instalados	<code>dpkg -l</code>	<code>rpm -qa</code>

1.3 Administrando escritorios

En todo lo que nos rodea y en todo lo que nos mueve, debemos advertir que interviene en algo la casualidad.

—Anatole France.

En el capítulo anterior, hemos hablado de la GCS y de cómo las distribuciones GNU/Linux utilizan el sistema de paquetería para garantizar la integridad frente al cambio.

Si tienes un equipo doméstico, todos los cambios producidos y liberados por los distintos proyectos y que hayan sido empaquetados y liberados por tu distribución GNU/Linux, serán instalados convenientemente con el simple hecho de dar la orden al gestor de paquetes para que actualice tu sistema.

Ahora bien, en una organización donde se requiera administrar los escritorios esto no es suficiente, veamos el porqué.

1.3.1 La personalización

La primera dificultad importante a la que se va a enfrentar un administrador, va a ser la de la personalización.

Imagina que tienes que migrar y administrar 1000 equipos a GNU/Linux y que tienes en tu red un servicio NTP, requiriéndose que todos tus escritorios estén con la hora sincronizada con este servicio.

Vas a tener que personalizar el cliente NTP en todos tus escritorios.

Una manera que se suele utilizar es instalar en un equipo una distribución GNU/Linux desde un DVD, editar el fichero de configuración del cliente NTP y configurar la IP (o el nombre DNS) del servidor donde se encuentra el servicio NTP. Después, puedes crear una imagen del disco duro con un sistema de clonado como pueda ser [Clonezilla](#) y clonar uno a uno los equipos usando dicha imagen.

Con este método, la personalización inicial reside en dicha imagen, pero sigamos imaginando. . .

Un día, a mitad de migración, recibes un correo y lees:

Alberto: El servicio NTP dejará de dar servicio a partir del día 10. En su lugar vamos a disponer de un nuevo servicio, llamado QueHoraEs, que es mucho mejor porque...

En este momento, ya estarás pensando en los 400 equipos que tienes migrados y te echarás las manos a la cabeza porque es evidente que este sistema de personalización no es adecuado.

Nota: La personalización inicial es muy sencilla de realizar, pero un cambio en la personalización puede darse en cualquier momento, y tienes que estar preparado para poder realizarlo.

Gestores de Sistemas

Afortunadamente, existen unas herramientas denominadas **Gestores de Sistemas** (*Systems Management Systems*), que pueden ayudarnos en la administración de los escritorios.

Algunos de estos Gestores de Sistemas se centran en la adquisición del estado de los equipos como *Nagios*, y otros permiten automatizar tareas mediante la ejecución de código en los equipos de manera centralizada como *Zenworks*, *Landscape*, *chef*, *puppet*, *cfengine*, *ansible*.

Los Gestores de Sistemas están muy influidos por las iniciativas realizadas en los sistemas de gestión de redes de telecomunicaciones, pudiendo realizar una o un conjunto de las siguientes tareas:

- Inventario hardware.
- Monitoreado de disponibilidad de servidores y mediciones.
- Inventario e instalación de software.
- Gestión de antivirus y anti-malware.
- Monitoreado de las actividades de los usuarios.
- Monitoreado de la capacidad de los sistemas.
- Gestión de seguridad.
- Gestión de almacenamiento.
- Monitoreado de la utilización y capacidades de la red.

Estas tareas podemos clasificarlas de acuerdo a **FCAPS**, un modelo y marco de trabajo de red de la gestión de telecomunicaciones de ISO para la gestión de redes. FCAPS es un acrónimo de Fault, Configuration, Accounting, Performance, Security (Falla, Configuración, Contabilidad, Desempeño, Seguridad), que son las categorías en las cuales el modelo ISO define las tareas de gestión de redes.

- **Fault** (Fallas): Es un evento que tiene un significado negativo. Su objetivo es reconocer, aislar, corregir y registrar fallos. Puede utilizar análisis de tendencias para predecir errores. Cuando se detecta un fallo o evento, se envía una notificación.
- **Configuration** (Configuración): En el proceso de gestión de la configuración, las operaciones diarias son monitoreadas y controladas.

Los objetivos de la gestión de la configuración son:

- Recolectar información.
- Modificar la configuración.
- Generación de reportes
- Gestión de cambios.

Los cambios de Hardware y Software son controlados por este proceso:

- Actualización, Instalación y eliminación de programas.
- Actualización, Instalación y eliminación de equipamiento (impresoras, scanners, memoria, etc.)

Este proceso debe tener en cuenta:

- Permitir acceso rápido a la información de la configuración.
 - Facilitar la configuración remota de los dispositivos.
 - Proporcionar un inventario actualizado de Software y Hardware.
 - Simplificación de la configuración de dispositivos.
 - El seguimiento de cambios a la configuración.
- **Accounting** (Contabilidad): Su objetivo es reunir las estadísticas de los usuarios.
 - **Performance** (Desempeño). Recolectando y analizando los datos de rendimiento, el estado general de los sistemas pueden ser monitorizado. Las tendencias pueden avisar de fallos de capacidad o de cuestiones relacionadas con la fiabilidad de los sistemas, antes de que en estos ocurran. Umbrales de rendimiento pueden ser establecidos para lanzar alarmas que serían controladas por la gestión de fallos habitual. Las alarmas se pueden clasificar atendiendo al grado de severidad.
 - **Security** (Seguridad). Se encarga de controlar el acceso a recursos de red. La seguridad de los datos puede ser conseguida con la autenticación, cifrado y permisos, principalmente.

Nota: Migasfree atendiendo a FCAPS, tiene capacidades de Faults, Configuration y Accounting.

Un ejemplo de funcionamiento típico de un Gestor de Sistemas que incorpore tareas de *Configuration* usaría un lenguaje que especificaría a qué estado se quiere llevar a los equipos, no cómo llegar a ese estado. En nuestro caso, sería algo parecido a esto:

- asegúrate de que el paquete ntp-client está desinstalado,
- asegúrate de que el paquete quehoraes-client está instalado,
- asegúrate de que el fichero de configuración de quehoraes-client es el mismo que el que está en el servidor.

Periódicamente, los clientes se conectarían al servidor para obtener este código que será ejecutado mediante el intérprete propio del Gestor de Sistemas instalado en el cliente.

Este sistema permite automatizar aquellas tareas que realizan a menudo los administradores de sistemas, y aunque algunos Gestores de Sistemas se las ingenian para llevar un control de versiones, mantienen una base de datos independiente a la de los backends de los gestores de paquetes, dejando en entredicho todo lo relativo a la integridad de los sistemas.

Empaquetando la personalización

En AZLinux usamos otro método: empaquetamos siempre la personalización.

Para el caso del cliente “QueHoraEs”, crearíamos el paquete azl-quehoraes-client⁴ con la siguiente información:

- Dependencias: quehoraes-client
- Obsoletos: ntp-client
- En el script de postinstalación escribiríamos el siguiente código:

En el fichero de configuración del cliente QueHoraes, modificar el valor de la entrada “server=” por la IP del servidor QueHoraEs

⁴ En AZLinux empleamos como nombre de paquete el prefijo «azl-» más el nombre del paquete que queremos personalizar.

¡Listo! Con esto queda garantizada la integridad frente al cambio de la personalización, aprovechándonos de la integridad que nos proporciona el sistema de paquetería de nuestra distribución GNU/Linux.

Una vez empaquetada nuestra personalización, se hace relativamente sencillo realizar cualquier cambio posterior en ella. Pero crear un paquete desde cero para personalizar una distribución GNU/Linux no es tan fácil, no tanto por la creación del paquete en sí, sino porque la personalización requiere de los conocimientos suficientes sobre el sistema GNU/Linux y sobre la propia aplicación que se personaliza.

Nota: Empaquetar la personalización nos asegura la integridad de los sistemas frente a sus cambios.

Date cuenta que no es necesario ningún Gestor de Sistemas para instalar dicha personalización. Sólo necesitas el Gestor de Paquetes, y éste siempre lo tienes disponible en cualquier distribución GNU/Linux.

Niveles de personalización

Las aplicaciones suelen incorporar dos niveles de personalización:

- La del usuario.
- La del sistema (para todos los usuarios del sistema).

La personalización del usuario es prioritaria a la del sistema, siempre y cuando esta última no sea obligatoria.

Es conveniente conocer si la aplicación que vas a configurar incorpora la personalización a nivel de sistema, ya que ésta es la que se tendrá que configurar.

En los casos en que las aplicaciones sólo tengan la configuración a nivel de usuario, o en los casos en los que se requiera, tendrás que recorrer todos los usuarios para aplicar la personalización a cada uno de ellos.

1.3.2 La liberación

Es el segundo problema importante con el que vas a tener que lidiar.

Por un lado, debes independizarte de los repositorios públicos de tu distribución GNU/Linux, por el simple motivo de que no puedes permitir que el control de los cambios que se instalarán en tus máquinas, lo tenga tu distribución GNU/Linux en vez de tu organización.

¿Imaginas que habría pasado en AZLinux cuando OpenSuSE sustituyó OpenOffice por LibreOffice? Cuando los usuarios hubieran encendido las máquinas a las 8:00 de la mañana, se iniciaría la actualización a LibreOffice automáticamente pudiéndose producir muchas incidencias. ¿Funcionaría todo? ¿No es mejor probar LibreOffice en tu organización antes de que se instale en todos tus equipos?

Tener la posibilidad de deshacer un cambio que se haya determinado como no deseado es importante.

Tienes que decidir por ti mismo el software que deben tener tus usuarios y, por tanto, debes tener los gestores de paquetes configurados contra tus propios repositorios de paquetes y gestionarlos de alguna manera.

Además, es conveniente que puedas planificar a quién y cuándo se deben liberar dichos cambios.

Imagina nuevamente el ejemplo de la sustitución de OpenOffice por LibreOffice. Estaríamos hablando de una actualización de cerca de 500 MB por equipo que, multiplicado por todos los equipos de una organización, podría resultar en mucho tráfico de red.

Una ventaja de planificar la liberación es que permite distribuir poco a poco los cambios, de tal manera que, si hay errores, afectará inicialmente a muy pocos equipos, permitiendo actuar de manera más relajada para corregir cualquier incidencia.

Por todo esto, y como los repositorios estándar de las distribuciones no tienen ningún mecanismo de planificación de la liberación, es por lo que decidimos desarrollar migasfree, extendiendo el concepto de repositorio de paquetes al concepto de despliegue: repositorio de paquetes dinámico y planificable.

1.3.3 Despliegue Migasfree

Un despliegue de migasfree es, simplemente, un repositorio estándar más la capacidad de poder especificar, de forma centralizada, cuándo y quién accede a ese repositorio.

Veamos como actúa migasfree en lo relativo a los repositorios:

1. Los cambios que se quieren liberar son empaquetados y subidos a un servidor migasfree.
2. Se crea un despliegue con los paquetes subidos y se establece a quién (atributos de usuario + equipo) y en qué momento se deben aplicar dichos cambios.
3. El servidor migasfree crea un repositorio físico (idéntico al de cualquier distribución GNU/Linux) con dichos paquetes, utilizando las herramientas estándar de creación de repositorios (`createrepo` para paquetería RPM o `dpkg-scanpackages` para paquetería Debian).
4. Cuando un cliente migasfree se conecta al servidor envía sus atributos al servidor.
5. El servidor consulta los despliegues para determinar, en función de esos atributos enviados, la lista de los repositorios físicos que tiene el cliente a su disposición y se los envía al cliente.
6. El cliente migasfree configura la lista de los repositorios físicos recibidos desde el servidor en el Gestor de Paquetes.
7. A continuación, el cliente migasfree da instrucciones al Gestor de Paquetes para que se produzca la eliminación, instalación y actualización de los paquetes desde los repositorios físicos.

1.3.4 La GCS en tu organización

En el capítulo anterior, hemos visto el proceso de la GCS en los distintos proyectos de software libre y también en las distribuciones GNU/Linux.

Pues bien, en una organización también debe realizarse el proceso de la GCS.

En AZLinux realizamos nuestra propia GCS y vemos como, de nuevo, se repiten las mismas actividades: petición de cambio, cambio y liberación.

Usamos dos tipos de peticiones de cambio:

- **Actualización de aplicaciones.** Si recibimos una petición para actualizar, por ejemplo, Mozilla Firefox, descargamos desde los repositorios de la distribución la versión deseada. La probamos en laboratorio, registrando cualquier información relevante en la petición de cambio. Finalmente, si todo es correcto, se liberan los paquetes a través de un despliegue migasfree, planificando su distribución (ver A en figura 3.2)
- **Personalización de aplicaciones.** Se produce cuando llega p.e., una petición de cambio para añadir un motor de búsqueda de sinónimos a Mozilla Firefox. Introducimos entonces en un paquete propio de AZLinux (`azl-firefox`), el código que instala dicho motor de búsqueda y liberamos dicho paquete en un despliegue de migasfree planificando su distribución (ver B en figura 3.2).

Las herramientas que usamos actualmente en cada actividad son:

- En la petición de cambio:
 - Gestor de proyectos: [Redmine](#)
- En el cambio:

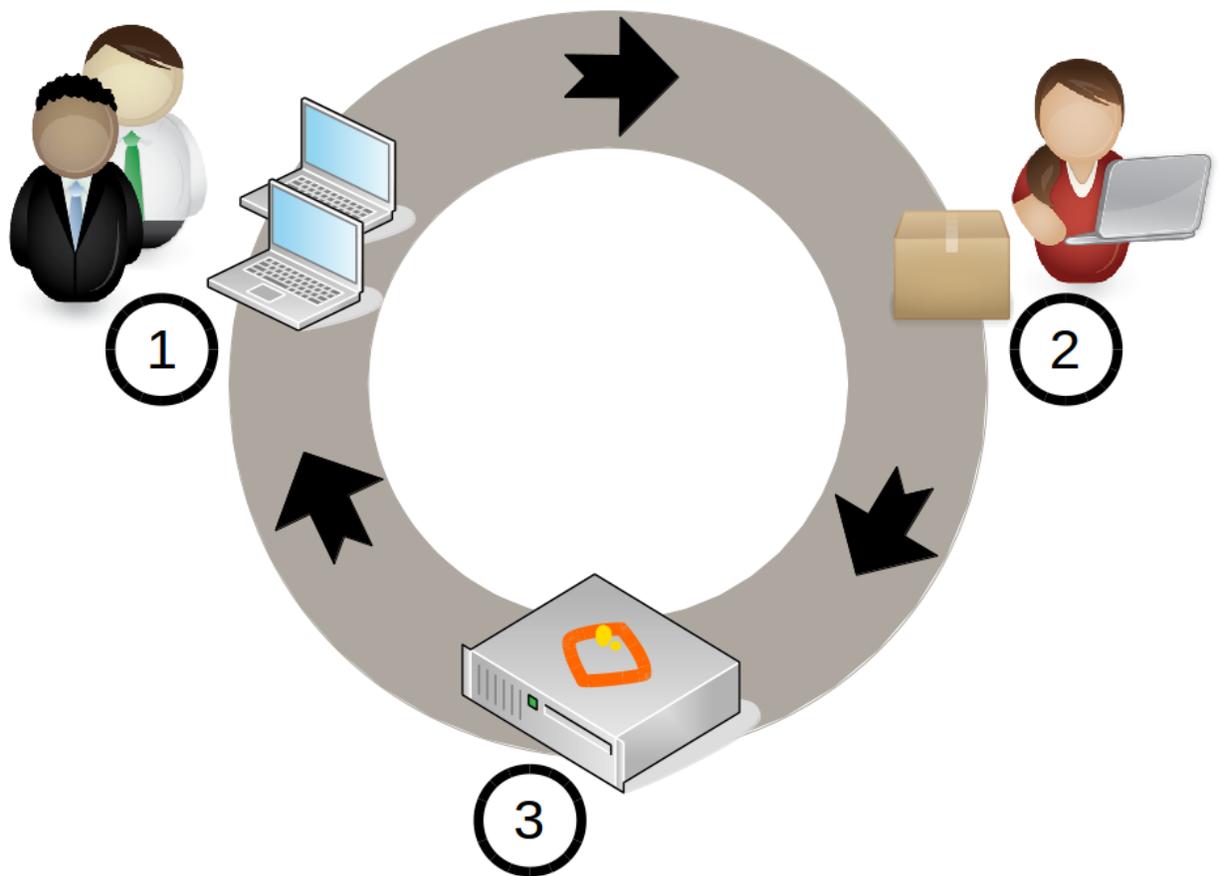


Figura 2: Proceso GCS en tu organización con migasfree.

1. Un usuario hace un **petición** de cambio.
2. Un desarrollador programa el **cambio** de la configuración software dentro de un paquete y lo sube a un servidor migasfree.
3. La **liberación** es realizada por el servidor migasfree a los ordenadores requeridos.

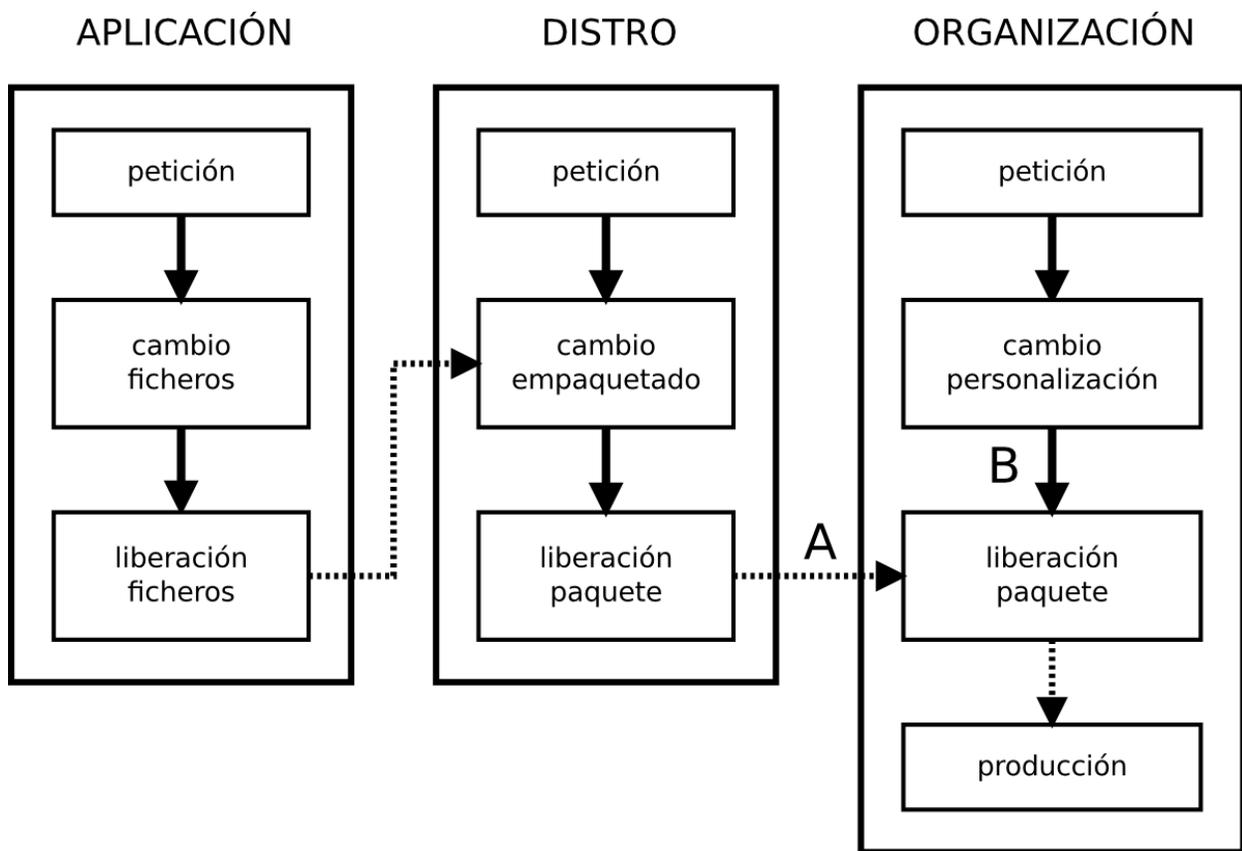


Figura 3: Procesos de la Gestión de la Configuración Software

- Editor de textos: [Geany](#)
- IDE: [PyCharm](#)
- Sistema de control de versiones: [GitLab](#)
- Gestor de proyectos: [Redmine](#)
- En la liberación:
 - Gestor de sistemas: [Migasfree](#)
 - Gestor de proyectos: [Redmine](#)

Nota: Migasfree nos proporciona, de manera centralizada, conocer el estado, no sólo del servidor migasfree, sino de cada uno de los equipos registrados en el servidor, convirtiéndose en una herramienta ideal para hacer una auditoría tanto de software como de hardware.

Beneficios

Los principales beneficios que obtendrá tu empresa, como resultado de aplicar una GCS, serían:

1. Reducción del coste de los servicios de desarrollo y mantenimiento.
2. Optimización del uso de los recursos.

Y para ti, como administrador:

1. Dispondrás de equipos más estables.
2. Vas a pasar de ser un administrador que se echa las manos a la cabeza ante cualquier cambio a ser un administrador favorecedor del cambio, ya que dispones de las herramientas para hacer el seguimiento y control de los cambios.
3. Y, en última instancia, vas a mejorar sustancialmente la resolución de incidencias.

1.4 Características de migasfree

Las cosas no se dicen, se hacen, porque al hacerlas se dicen solas.

—Woody Allen.

1.4.1 El nacimiento de migasfree

En el año 2005, todos grupos políticos del Ayuntamiento de Zaragoza manifestaron por unanimidad, en pleno de gobierno municipal, apoyar las políticas de uso de Software Libre y, en concreto, el fomento de los programas de SL en el entorno de escritorio del empleado municipal. La Dirección General de Ciencia y Tecnología asume, inicia y potencia este importante reto.⁵

Este proyecto se planificó en tres etapas:

- Primera: Migrar a aplicaciones que presentaban un impacto bajo sobre usuarios y técnicos en el Sistema Operativo actual (por aquel entonces, Microsoft Windows XP).
- Segunda: Migrar la plataforma ofimática Microsoft Office 97 por la suite libre OpenOffice.

⁵ Eduardo Romero Moreno, *Migración Escritorio Software Libre*, 2011

- Tercera: Sustituir el SO Windows XP por un sistema operativo basado en Linux. Esta etapa se inició en 2008 y todavía sigue abierta.

Para iniciar la tercera etapa, se tuvieron que realizar los primeros prototipos de lo que llegaría a ser la primera versión de AZLinux. En estos prototipos, la personalización se realizaba manualmente, en un equipo cuya imagen del disco duro nos servía para clonarla en otros equipos y hacer las pertinentes pruebas.

En aquel tiempo, aprendimos a empaquetar y empezamos a introducir nuestra personalización en nuestros propios paquetes. La ventaja frente a la personalización manual era muy significativa.

Con los primeras migraciones reales, nos surgió la necesidad de actualizar nuestros paquetes y, después de probar sin éxito *Zenworks for Linux*, decidimos crear nuestros propios repositorios de paquetes. Quisimos emular lo que ya estábamos haciendo con los escritorios XP, esto es, distribuir software basándonos en el contexto al que pertenecía un usuario en nuestro LDAP. Con un poco de *Bash scripting*, en Mayo de 2009, implementamos lo que serían unos repositorios dinámicos que se configuraban en el cliente en función del contexto.

Esto fue, sin duda, una gran idea, pero la gestión de estos repositorios dinámicos era manual y muy propensa a errores.

La gestión de estos repositorios dinámicos recayó en mí, por lo que decidí simplificarla inmediatamente y crear el primer prototipo de *migasfree*. Dos semanas de programación, en horas no laborales, fueron suficientes para presentar a mis compañeros de trabajo un prototipo, que fue puesto en producción en Junio de 2009.

Nota: Una de las ventajas de trabajar con software libre es la facilidad con la que puedes crear proyectos ya que puedes mezclar, como si de piezas de puzzle fueran, diferentes componentes sin preocuparte en exceso del tema de las licencias. Un ejemplo de esto ha sido la incorporación de la funcionalidad de captura del hardware en los equipos. Utilicé el comando `lshw` y unas pocas líneas de código para adaptarlo a la base de datos de *migasfree*.

1.4.2 Versiones

El primer prototipo sólo trabajaba con paquetería `rpm` y gestor de paquetes `yum`, y el código `bash` que se ejecutaba en el cliente se generaba en el servidor.

Después de usar *migasfree* un tiempo en producción, vimos que podría ser un buen sistema para otras organizaciones, y mis compañeros me dieron el impulso necesario para publicar el código. Así, durante el verano de 2009, reorganice los menús, limpié un poco el código, e hice que *migasfree* pudiera trabajar con distintas versiones de SO y de sistemas de paquetería. Fue publicado en [github](#) en abril de 2010 y bautizado como «*migasfree with fried eggs*», porque mis compañeros decían que el logotipo se parecía a un huevo frito. ¡Qué sabrán ellos de Arte!

En Noviembre de 2011, Jose Antonio Chavarría, desarrollador de AZLinux, reescribe y publica el [cliente migasfree](#). Realizó también grandes cambios en la estructura del servidor. Tuvimos que definir la API con la que el cliente y el servidor debían comunicarse. Usamos claves asimétricas para dotar de seguridad al sistema. Esta nueva versión fue denominada «*migasfree no trans*» supongo que por incorporar un código más «limpio», por decirlo de alguna manera.

Poco a poco, fuimos dotando al sistema de nuevas funcionalidades, y para principios de 2013, Jose Antonio Chavarría cambió la navegación y aspecto del servidor. Esta nueva versión fue denominada «*migasfree with chocolate*».

En febrero de 2014, liberamos la versión 4 del servidor (*migasfree grape edition*). Esta versión hace uso de [bootstrap](#) con el fin de dotar a la aplicación de un diseño web adaptable a distintos dispositivos. Además, incorpora distintas mejoras de todo tipo. Actualmente es la versión que utilizamos en AZLinux.

En 2015 nos sumamos a una ola disruptora: [docker](#). Esto nos ha liberado de tener que pelearnos con las dependencias de los componentes que utilizamos en el servidor y de tener que publicar los paquetes que generábamos para distintas distribuciones GNU/Linux (utilizamos versiones de componentes que aún no han sido liberados y esto nos creaba auténticos quebraderos de cabeza).

Con *docker* hemos conseguido colocar el servidor y sus dependencias en un contenedor virtual (un *debian*) que puede ejecutarse en cualquier servidor GNU/Linux. Esto nos ha proporcionado una flexibilidad y portabilidad para ejecutar el servidor de una forma muy simple tanto en un equipo físico como en la nube.

1.4.3 Características

- Migasfree es simple, y hacemos esfuerzos por mantenerlo así. Tendemos a lo que denominamos gestión cero, es decir, procuramos que la gestión de añadir nuevas entradas en migasfree no requiera ninguna tarea administrativa.
- Está basado en la arquitectura cliente/servidor.
- Es seguro. Las comunicaciones entre cliente y servidor están firmadas con claves asimétricas.
- Es adaptable. Puedes programar tus propias fórmulas para obtener los atributos de los ordenadores y usuarios según tus intereses.
- Es Software Libre licenciado bajo la **GNU Public License**.
- Almacena tanto el inventario software y hardware de los equipos, permitiendo hacer consultas sobre ellos. Almacena también información de los equipos tales como sus atributos, sincronizaciones, migraciones que se han realizado, etc.
- Consultas. Puedes programar consultas contra la base de datos de migasfree.
- Gestión de errores. Los errores que se producen en los equipos son enviados al servidor y almacenados, permitiendo hacer su seguimiento.
- Gestión de fallas. Puedes programar código que será ejecutado en los clientes con el fin de obtener información de los equipos.
- Alertas. Permite conocer en tiempo real el estado del sistema facilitando al administrador su trabajo.
- Estadísticas.

1.4.4 Principales componentes empleados

- Lenguaje de programación *Python*.
- *Django* un framework de desarrollo web.
- Servidor web *Nginx*.
- *Chaussette* como servidor WSGI
- Base de datos *Postgresql*.
- Información Hardware: *Lshw*.

La indeterminación de la chuleta de cerdo.

2.1 Probando migasfree

La unidad es la variedad, y la variedad en la unidad es la ley suprema del universo.

—Isaac Newton.

El objetivo de este capítulo es que dispongas rápidamente de un servidor y un cliente *migasfree* totalmente funcional, por eso no me voy a extender en explicaciones.

Nota: Usa una máquina virtual de [virtualbox](#) realizando la instalación mínima por red de [Debian 64 bits](#) para ver el funcionamiento de *migasfree* y familiarizarte con él antes de poner a *Migasfree en producción*

2.1.1 Instalando el servidor

Si ahora no quieres preocuparte de cómo se realiza esta instalación, nunca has usado [docker](#) o, simplemente, quieres avanzar más rápido, te proporcionamos este [Open Virtual Appliance \(OVA\)](#) para que lo ejecutes con *virtualbox*. Contiene un servidor *migasfree* ya instalado.

Con *virtualbox* instalado, y una vez descargado el fichero OVA, haz doble click en él y, a continuación, observa las siguientes reglas de reenvío de puertos accediendo al menú de *virtualbox*: configuración-red-avanzadas-reenvío de puertos:

Nombre	Protocolo	IP anfitrión	Puerto	IP invitado	Puerto invitado
=====	=====	=====	=====	=====	=====
Rule 1	TCP	127.0.0.1	2222	10.0.2.15	22
Rule 2	TCP	127.0.0.1	8080	10.0.2.15	80

Ya puedes arrancar la máquina virtual.

Los usuarios y contraseñas de esta máquina virtual son: tux:tux y root:root.

Para acceder por SSH a esta máquina virtual usa:

```
ssh -p 2222 tux@127.0.0.1
```

Una vez dentro, puedes loguearte como `root` de la siguiente manera:

```
tux@migasfree:~$ su
```

Para acceder al servidor web instalado en esta máquina virtual, usa la URL <http://127.0.0.1:8080> desde el equipo anfitrión.

Ahora bien, si quieres hacer la instalación tú mismo, accede a la máquina debian 64 bits sobre la que vas a instalar el servidor y sigue los pasos indicados en [migasfree-docker](#).

2.1.2 Comprobando el servidor

En un navegador web accede a la URL del servidor. Si todo ha ido bien, verás la figura 5.1.

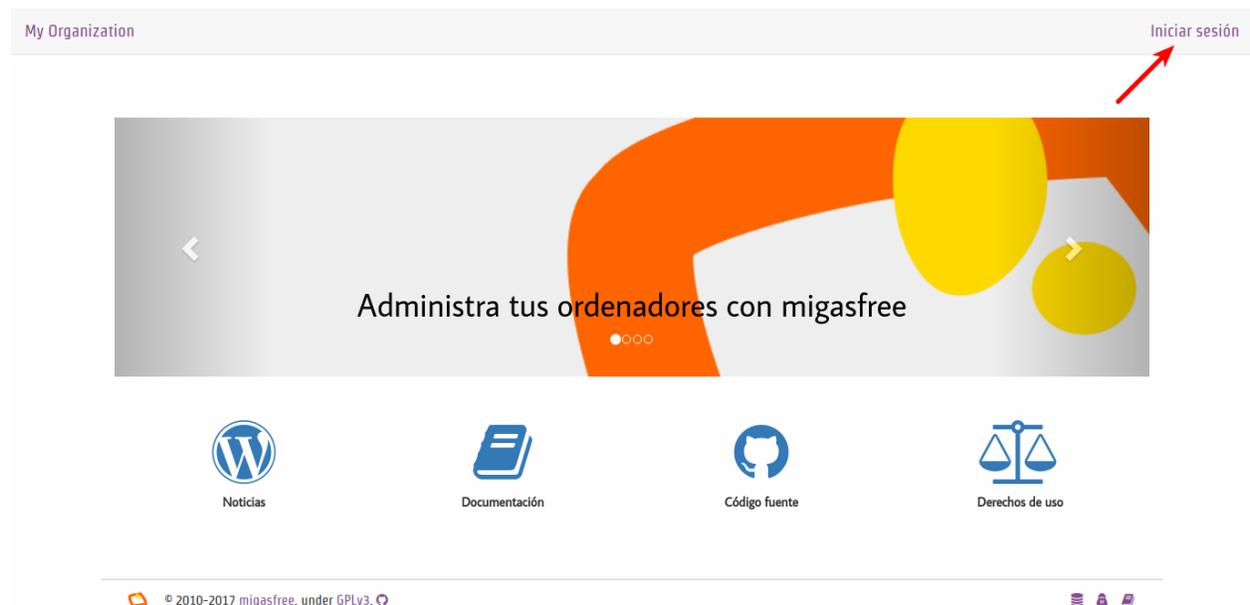


Figura 1: Acceso al servidor migasfree.

Pulsa en `iniciar sesión` y haz login con el *nombre de usuario* «admin» y *contraseña* «admin». Verás algo parecido a la figura 5.2. Observa como arriba a la derecha pone `alertas 0`. Esto nos indica que todo está controlado.

2.1.3 Instalando el cliente

Instalando el paquete migasfree-client

Ahora instala el cliente *migasfree* sobre la misma máquina donde has instalado el servidor. Para ello, actualiza la lista de paquetes e instala el paquete `migasfree-client`:

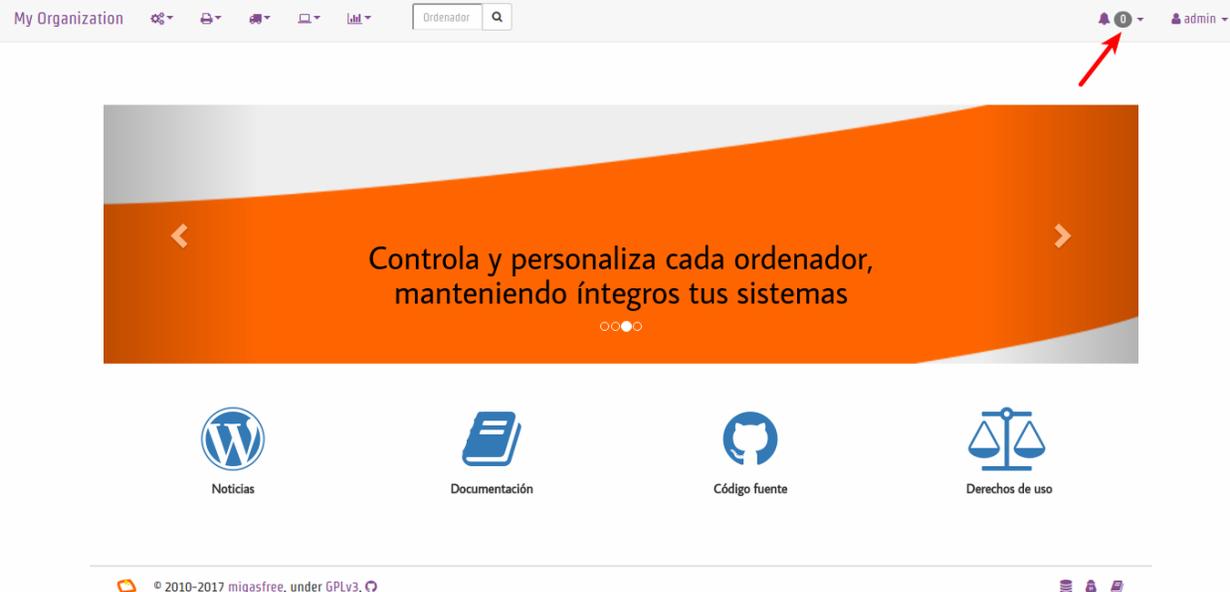


Figura 2: Estado del servidor con 0 alertas.

```
# wget -O - http://migasfree.org/pub/install-client | bash
```

También puedes instalar el cliente en cualquier otra máquina y editar el fichero `/etc/migasfree.conf` manualmente para configurarlo. Descomenta la línea `# Server = localhost` y asígnale la dirección del servidor web. Te hago notar que esta **no es la manera** en que debemos hacerlo, pero por ahora puede servirnos. La manera correcta sería empaquetar dicha modificación, cosa que veremos más adelante en *Configurando migasfree-client*

Nota: Cualquier cambio en la configuración de las aplicaciones o del S.O se podrán realizar de manera centralizada con suma facilidad, manteniendo además la integridad, sólo si dicha configuración ha sido empaquetada.

Registrando el cliente

Ejecuta el comando:

```
# migasfree -u
```

te devolverá una salida parecida a esta:

```
root@migasfree:/home/tux# migasfree -u
Sesión gráfica no detectada
Versión de migasfree client: 4.15

Opciones de ejecución: /etc/migasfree.conf
Proyecto: debian-9.3
Servidor: localhost
Actualizar paquetes automáticamente: True
```

(continues on next page)

```
Proxy: None
Certificado SSL: None
Proxy caché de paquetes: None
Depuración: False
Nombre del ordenador: migasfree
GUI detallado: True
PMS: apt-get

Usuario gráfico: root

Autoregistrando ordenador...
;Clave /var/migasfree-client/keys/localhost/debian-9.3.pri creada!
;Clave /var/migasfree-client/keys/localhost/server.pub creada!
Warning: apt-key output should not be parsed (stdout is not a terminal)
;Clave /var/migasfree-client/keys/localhost/repositories.pub creada!

***** Conectando al servidor migasfree... *****
***** Correcto

***** Obteniendo propiedades... *****
***** Correcto

***** Evaluando atributos... *****
PCI: 8086:1237~Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02) ,
↳8086:7000~ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II] ,
↳8086:7111~IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01) ,
↳80ee:beef~VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics
↳Adapter ,8086:100e~Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet
↳Controller (rev 02) ,80ee:cafe~System peripheral: InnoTek Systemberatung GmbH
↳VirtualBox Guest Service ,106b:003f~USB controller: Apple Inc. KeyLargo/Intrepid USB
↳ ,8086:7113~Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08) ,8086:2829~
↳SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI
↳mode] (rev 02) ,

HST: migasfree

NET: 10.0.2.0/24

***** Subiendo atributos... *****
***** Correcto

***** Ejecutando fallas... *****
LOW_HOME_PARTITION_SPACE:
LOW_SYSTEM_PARTITION_SPACE:

***** Subiendo fallas... *****
***** Correcto

***** Creando repositorios... *****
***** Correcto

***** Obteniendo los metadatos de los repositorios... *****
Ign:1 http://ftp.es.debian.org/debian stretch InRelease
Obj:2 http://security.debian.org/debian-security stretch/updates InRelease
Obj:3 http://ftp.es.debian.org/debian stretch-updates InRelease
Obj:4 http://ftp.es.debian.org/debian stretch Release
Obj:5 https://download.docker.com/linux/debian stretch InRelease
Leyendo lista de paquetes... Hecho
***** Correcto

***** Desinstalando paquetes... *****
***** Correcto

***** Instalando paquetes obligatorios... *****
***** Correcto
```

(proviene de la página anterior)

```
***** Actualizando paquetes... *****
DEBIAN_FRONTEND=noninteractive /usr/bin/apt-get -o APT::Get::Purge=true -o
↳Dpkg::Options::=--force-confdef -o Dpkg::Options::=--force-confold -o
↳Debug::pkgProblemResolver=1 --assume-yes --force-yes --allow-unauthenticated --auto-
↳remove dist-upgrade
Leyendo lista de paquetes...
Creando árbol de dependencias...
Leyendo la información de estado...
Calculando la actualización...
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.

***** Correcto

***** Subiendo el inventario del software... *****
***** Correcto

***** Capturando información sobre el hardware... *****
***** Correcto

***** Enviando información sobre el hardware... *****
***** Correcto

***** Operaciones completadas *****
```

2.1.4 Comprobando el estado del servidor

Comprueba los datos que se han recogido accediendo al servidor con tu navegador web.

- Fíjate ahora que tienes 2 Alertas (figura 5.3). Pulsa sobre ellas y luego

sobre 2 alertas por comprobar:

- La primera te notifica que el ordenador CID-1 ha dado de alta la plataforma Linux.
- La segunda notificación te dice que el ordenador CID-1 ha añadido el proyecto debian-x.x.

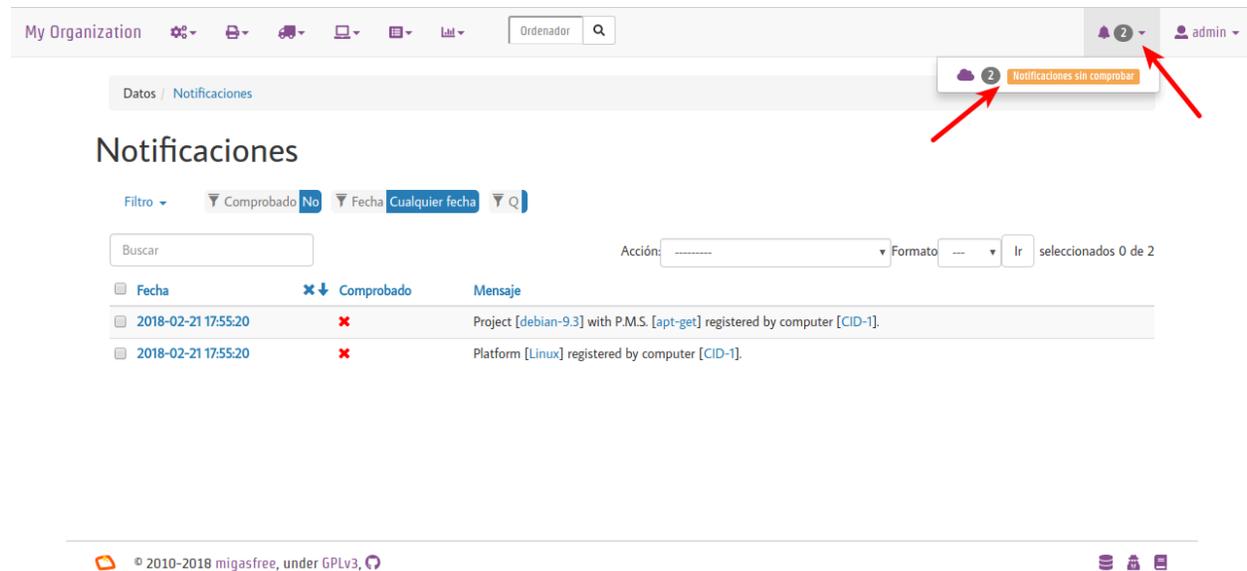


Figura 3: Notificaciones.

- Accede a Datos – Ordenadores y observa: (figura 5.4)
 - Los datos del ordenador CID-1 (pulsando sobre CID-1).
 - Su hardware.

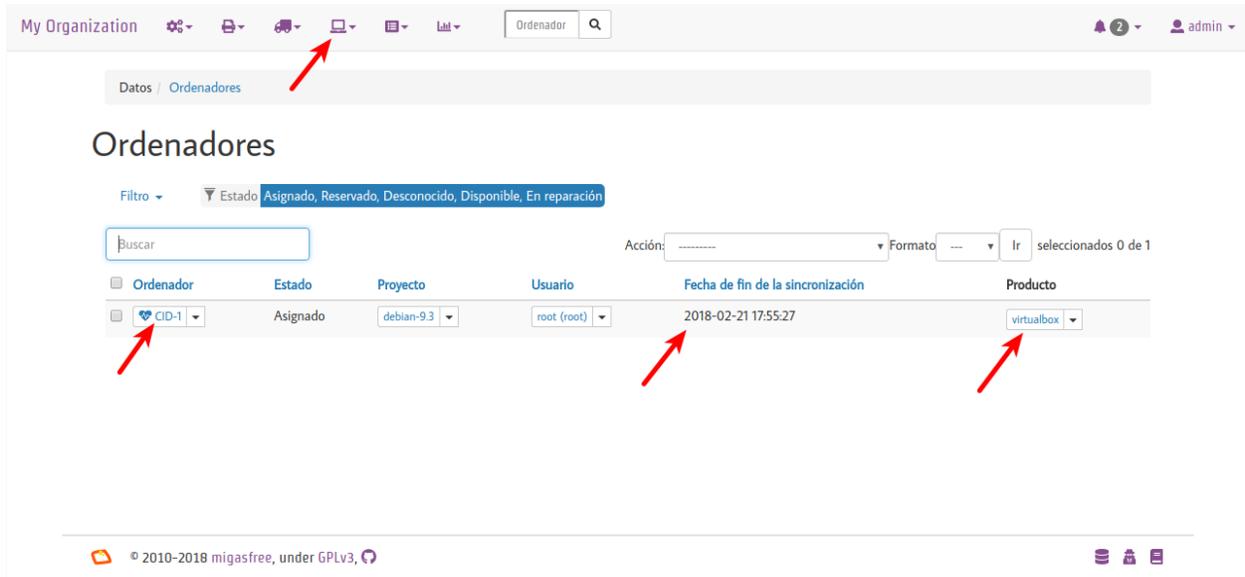


Figura 4: Ordenadores.

Nota: CID es un acrónimo de Computer IDentificator. Es un número que el servidor asigna a cada ordenador para identificarlo. Es imposible tener dos ordenadores con el mismo CID.

Nota: El símbolo que aparece a la izquierda del CID (el corazón) indica el estado en el que encuentra el ordenador.

¡Enhorabuena! Has instalado un servidor migasfree y has registrado en él tu primer ordenador.

2.1.5 Desplegando software

Y ahora, para ir abriendo boca, vamos a instalar y eliminar aplicaciones de manera centralizada usando el servidor *migasfree*.

Supón que quieres sustituir `nano` por `vim` en todos los equipos de tu organización.

Accede a Liberación – Despliegues y pulsa en en símbolo + para añadir un despliegue.

Nota: Sitúa el cursor sobre los iconos del formulario de despliegue para identificar cada campo.

Introduce los siguientes datos:

- Nombre: sustituir `nano` por `vim`
- proyecto: (el que corresponda)

- paquetes a instalar: vim
- paquetes a desinstalar: nano
- atributos incluidos: ALL SYSTEMS

Graba el despliegue.

Ahora sincroniza el equipo cliente con el servidor *migasfree*:

```
# migasfree -u
```

Puedes comprobar en la salida estándar del cliente *migasfree* que se ha configurado el repositorio *sustituir-nano-por-vim* y que se ha instalado el paquete *vim* y desinstalado *nano*.

```
***** Obteniendo los metadatos de los repositorios... *****
Obj http://localhost sustituir-nano-por-vim InRelease
Des:1 http://localhost sustituir-nano-por-vim/PKGS amd64 Packages [29 B]

...
***** Desinstalando paquetes... *****
Los siguientes paquetes se ELIMINARÁN:
  nano*
***** Correcto

***** Instalando paquetes obligatorios... *****
Se instalarán los siguientes paquetes NUEVOS:
  vim
***** Correcto
```

Nota: En el despliegue que acabamos de hacer sólo hemos dado la orden de instalar y desinstalar paquetes, pero además se ha creado un repositorio de paquetes vacío. En los siguientes capítulos aprenderás a subir tus propios paquetes al servidor e introducirlos en los despliegues, pero vayamos despacito, suave, suavecito.

Al poner el atributo `SET-ALL SYSTEMS` estamos indicando que se aplique este despliegue a todos los ordenadores. Podríamos haber incluido otros atributos como `CID-1`, `NET-10.0.2.0/24`, o una lista de ellos. Sólo se aplicará este despliegue a los ordenadores que tengan algún atributo coincidente con los atributos incluidos en el despliegue.

Observa que también puedes excluir ordenadores. El servidor sigue la siguiente lógica: primero comprueba los ordenadores incluidos y después excluye los ordenadores que tenga algún atributo que coincida con los atributos excluidos del despliegue.

Nota: Ahora cada vez que un ordenador se sincroniza con el servidor (y se le aplica este despliegue), se instala *vim* y desinstala *nano* si es necesario.

Nota: Si ahora queremos instalar *vim* y *emacs* en todos los ordenadores, pon sus nombres en paquetes a instalar y deja en blanco paquetes a desinstalar:

- paquetes a instalar: vim emacs
 - paquetes a desinstalar:
-

Conforme los equipos se vayan sincronizando, se producirán los cambios.

Aprenderás más sobre los despliegues en el capítulo dedicado a *La Liberación*.

¡Enhorabuena de nuevo! Ya sabes como instalar/desinstalar software de manera centralizada a un conjunto de ordenadores. No está nada mal para empezar, ¿no crees?

En el siguiente capítulo vas a aprender a hacer el cambio de configuración software al estilo *migasfree*.

2.2 Configurando software al estilo migasfree

No esperes hasta que las condiciones sean perfectas para comenzar, el empezar hace las condiciones perfectas.

—Alan Cohen.

El objetivo de este capítulo es que veas todo el proceso de la **Gestión de la Configuración Software en conjunto**. Lo vamos a hacer desplegando un paquete que, simplemente, instalará unos ficheros de ejemplo (png, gif, mp4, etc.) que pueden servir al personal del Centro de Asistencia Usuarios (CAU) para comprobar las asociaciones de archivos y aplicaciones.

2.2.1 Al estilo tradicional

Imagina que te llega una petición de cambio para añadir en todos los ordenadores de escritorio de tu organización unos determinados ficheros de *test* para facilitar el trabajo de los compañeros del CAU.

Como aún no has pensado cómo desplegar software en los escritorios GNU/Linux, decides acceder uno a uno a los equipos por SSH y copiarlos manualmente. Tardas unos días pero al final realizas el trabajo justo antes de cogerte las vacaciones de verano, a excepción de algunos equipos que estaban apagados y que no has podido acceder remotamente.

Ahora bien, mientras estás de vacaciones, ¿podría responder fácilmente a las cuestiones siguientes tu compañero de trabajo?

- ¿Qué cambios se han realizado en un determinado equipo desde el 1 de Marzo?
- ¿Quién realizó el cambio?
- ¿Cuándo se desplegaron todos esos cambios en los equipos?
- ¿Qué equipos tienen un cambio determinado?

Difícilmente tu compañero va a poder reponder estas cuestiones de manera eficaz, aunque hayas registrado muy bien tu trabajo.

Podrías haberte ahorrado mucho trabajo usando herramientas como *engine*, *puppet*, *chef*, etc., para el despliegue de estos ficheros, pero aún así no podrías contestar fácilmente a dichas preguntas.

La integridad frente al cambio no está garantizada con este método.

A continuación, te propongo otra forma de realizar los cambios de configuración. Se basa en utilizar el empaquetado para trasladar los cambios a los equipos.

Nota: Usa la rueda, no la reinventes. Usando el sistema de paquetería para desplegar el software y su configuración, nos proporciona integridad frente a los cambios. Cualquier distribución GNU/Linux tiene un gestor de paquetes y hace muy bien su trabajo.

Asumo que tienes un gestor de proyectos, como *Redmine*, donde vas a registrar las peticiones de cambio (o al menos que hagas como que lo tienes) y que has completado con éxito el capítulo anterior. Todos los comandos de este capítulo los vas a ejecutar como **root** en el equipo que hayas utilizado en el capítulo anterior.

2.2.2 Tu primer cambio de configuración

El primer cambio sobre un **Elemento de Configuración Software** (ECS) es el que te llevará más trabajo, porque exige la creación de un paquete.

Petición

Imagina que te llega la siguiente la petición de cambio, que registras y aceptas en el gestor de proyectos:

Gestor de proyectos:

Registro: Copiar distintos ficheros de ejemplos: png, gif, jpg, etc., en todos los ordenadores para uso y disfrute de los compañeros del CAU.

Lo primero que haces es identificar al ECS que afecta, es decir, cuál es el paquete que debe ser modificado. Como no existe todavía un paquete sobre el que actuar, asigna la petición de cambio a un desarrollador (¡qué suerte, siempre te toca a ti!) y registra en la petición de cambio:

Gestor de proyectos:

Registro: Crear el paquete **acme-test-files**

Asignado a: *desarrollador*.

Cambio

Empaquetado

Como desarrollador, tienes que crear el paquete de configuración `acme-test-files`. Si nunca has creado un paquete, no te preocupes, para facilitarte las cosas y que puedas avanzar, centrándote en el proceso GCS, descárgate el proyecto `fun-with-migasfree-examples` donde se incluyen los ejemplos utilizados es este libro.

```
# apt-get install unzip
$ wget https://github.com/migasfree/fun-with-migasfree-examples/archive/master.zip
$ unzip master.zip
$ cd fun-with-migasfree-examples-master
```

Observa los ficheros que incluimos en el paquete `acme-test-files`:

```
$ ll acme-test-files/usr/share/acme-test-files/
```

Ya tienes el fuente del paquete. Ahora genera el paquete. Para ello, debes tener instalados algunos paquetes:

```
# apt-get install devscripts gcc build-essential:native
```

Y, ahora sí, genera el paquete:

```
$ cd acme-test-files
$ /usr/bin/debuild --no-tgz-check -us -uc
$ cd ..
```

¡Felicidades, el cambio está empaquetado en `acme-test-files_1.0-1_all.deb`!

Subiendo el cambio al servidor

Usa este comando para subir el paquete generado al servidor.

```
# migasfree-upload -f acme-test-files_1.0-1_all.deb
```

- Introduce usuario: admin
- Contraseña: admin
- Proyecto: debian-x.x (En mi caso la versión es debian-8.7)
- Almacén: acme

Finalmente, asigna la petición de cambio a un liberador (sí, otra vez vas a ser tú) y registra en la petición:

Gestor de proyectos

Registro: Creado paquete **acme-test-files_1.0-1_all.deb**

Asignado a: *liberador*

¡Felicidades! Has realizado un cambio de configuración y lo has almacenado en el servidor *migasfree*.

Liberación

Ahora vas a ver el punto de vista del encargado de liberar los cambios:

Accede mediante un navegador web a tu servidor. Observa que en Alertas tienes 1 paquete huérfano (Figura 6.2).

The screenshot shows the 'Paquetes/Conjuntos' (Packages/Sets) view in the Migasfree web interface. At the top, there is a navigation bar with 'My Organization', various icons, a search box, and a user profile 'admin'. A notification dropdown menu is open, showing '1 Paquetes/Conjuntos huérfanos' and '2 Notificaciones sin comprobar'. The main content area has a search bar, a table with columns for 'Paquete/Conjunto', 'Proyecto', 'Almacén', and 'Despliegues'. The table contains one entry: 'acme-test-files_1.0-1_all.deb' under 'Paquete/Conjunto', 'debian-8.10' under 'Proyecto', and 'acme' under 'Almacén'. The footer shows the copyright '© 2010-2018 migasfree, under GPLv3' and some icons.

Figura 5: Paquetes huérfanos.

Nota: Todos los paquetes que se han subido al servidor y no están asignados en ningún despliegue se denominan huérfanos.

Liberando el cambio de configuración

Ahora, vas a liberar el cambio creando un nuevo *despliegue*.

Para ello, ve a Liberación (pulsando sobre el icono del camión) y accede a Despliegues. Luego pulsa en el botón + para añadir un nuevo despliegue e introduce estos datos:

Nota: Coloca el cursor sobre los iconos para familiarizarte con el nombre de los campos del formulario.

- Nombre = ficheros de ejemplo #24543 (Una buena idea es hacer aquí una referencia al nº de petición del Gestor de proyectos)
- Proyecto = debian-x.x
- Paquetes disponibles = acme-test-files_1.0-1_all.deb
En este campo se asignan los paquetes que contendrá el repositorio físico asociado al despliegue.
- Paquetes a instalar = acme-test-files
En este campo se escriben los **nombres** de los paquetes que se instalarán **obligatoriamente** en los clientes.
- Atributos incluidos = SET-ALL SYSTEMS
De esta manera indicamos que todos los clientes tendrán acceso a este despliegue.

Guarda el despliegue.

Observa que, en Alertas, ya no tienes ningún paquete huérfano.

Registra y cierra la petición de cambio:

Gestor de proyectos

Registro: Liberado en Despliegue **ficheros de ejemplo #24543**.

Petición: *cerrada*.

Aplicando el cambio

Para aplicar el cambio a un equipo, ejecuta el siguiente comando:

```
# migasfree -u
```

Observa en la salida del comando:

```
***** Subiendo el historial del software... *****  
Diferencia en el software: # 2017-02-03 18:21:17  
+acme-test-files-1.0-1  
***** Correcto
```

Puedes comprobar que en /usr/share/acme-test-files tienes los ficheros de ejemplos incluidos en el paquete.

```
$ ls -la /usr/share/acme-test-files
```

2.2.3 Tu segundo cambio de configuración

Petición

Te llega una segunda petición de cambio, ya que a tus compañeros del CAU les ha venido bien estos ficheros de ejemplos pero echan de menos un SVG:

Gestor de proyectos

Registro: Añadir un fichero SVG de prueba para el CAU.

Como siempre, identificas primero el ECS al que afecta el cambio: En este caso es a `acme-test-files`. En la petición de cambio, asignas al desarrollador y registras:

Gestor de proyectos

Registro: Modificar el paquete `acme-test-file` añadiendo el SVG de ejemplo: <https://commons.wikimedia.org/wiki/File:Tux.svg>

Asignado a : *desarrollador*.

Cambio

Los cambios que se realizan sobre un paquete ya creado suelen ser más sencillos de realizar porque, simplemente, se modifica el paquete.

Empaquetado

Descarga el fichero SVG de <https://commons.wikimedia.org/wiki/File:Tux.svg> y cópialo en `acme-test-files/usr/share/acme-test-files`.

Edita el fichero del paquete `acme-test-files/debian/changelog` para registrar el cambio realizado. Tendrás que **añadir** estas líneas **al principio del fichero**:

```
acme-test-files (1.0-2) unstable; urgency=low

 * Added file tux.svg

-- Alberto Gacías <alberto@migasfree.org>  Fri, 3 Feb 2018 18:25:00 +0100
```

Presta atención a:

- La versión del paquete (**1.0-2**).
- Sustituir **tu nombre y dirección de correo**.
- Modificar la **fecha y hora**.

Nota: El formato que se utiliza en el **changelog** en paquetes debian es muy estricto. Ten cuidado con los espacios, retornos de carro y fechas.

Un aspecto que no hay que descuidar es el tema del *copyright* y licencia. Edita el fichero `acme-test-files/debian/copyright` y añade el *copyright* y licencia del fichero `tux.svg`

```
Files: tux.svg
Copyright: lewing@isc.tamu.edu Larry Ewing and The GIMP
License: https://creativecommons.org/publicdomain/zero/1.0/legalcode
```

Ahora, generamos el paquete:

```
$ cd acme-test-files
$ /usr/bin/debuild --no-tgz-check -us -uc
$ cd ..
```

Observa que se ha generado el mismo paquete, pero con la versión 1.0-2:

```
# root@debian8:~# ls -la *.deb
-rw-r--r-- 1 root root 2338 feb  3 17:49 acme-test-files_1.0-1_all.deb
-rw-r--r-- 1 root root 2398 feb  3 18:27 acme-test-files_1.0-2_all.deb
```

Subiendo al servidor el cambio

```
# migasfree-upload -f acme-test-files_1.0-2_all.deb
```

- Introduce usuario: admin
- Contraseña: admin
- Proyecto: debian-x.x
- Almacén: acme

Gestor de proyectos

Registro: Creado paquete **acme-test-files_1.0-2_all.deb**

Asignado a: *liberador*

Liberación

Liberando el cambio de configuración

Observa como aparece de nuevo un paquete huérfano en alertas y que corresponde a `acme-test-files_1.0-2_all.deb`.

Accede a Liberación - Despliegues y edita el despliegue ficheros de ejemplo #24543.

Añade a Paquetes disponibles el paquete `acme-test-files_1.0-2_all.deb`.

Guarda el despliegue.

Registra y cierra la petición de cambio:

Gestor de proyectos

Registro: Liberado **acme-test-files_1.0-2_all.deb** en Despliegue **ficheros de ejemplo #24543**.

Petición: *cerrada*.

Aplicando el cambio

Ejecuta de nuevo:

```
# migasfree -u
```

Observa en la salida de este comando el cambio de software:

```
***** Subiendo el historial del software... *****
Diferencia en el software: # 2017-02-03 18:30:21
+acme-test-files-1.0-2
-acme-test-files-1.0-1
***** Correcto
```

Comprueba si el cambio se ha aplicado.

```
$ ls -la /usr/share/acme-test-files
```

2.2.4 Auditoría

Ahora tu compañero sí que podría responder las siguientes cuestiones de manera centralizada desde el servidor migas-free aunque todos los equipos estén apagados y tú de vacaciones:

¿Qué cambios se han producido en el ordenador 1 y cuándo?

Accede a Datos - Ordenadores, accede al equipo CID-1 (CID=Computer ID) y mira el final del campo historial de software de la sección Software:

```
# 2017-02-03 18:21:17
+acme-test-files-1.0-1

# 2017-02-03 18:30:21
+acme-test-files-1.0-2
-acme-test-files-1.0-1
```

El signo (-) indica paquete desinstalado y el signo (+) paquete instalado.

¿Qué se cambió, quién y cuándo hizo el cambio?

Esta información está en el paquete como metainformación. Para acceder a ella, accede a Liberación - Paquetes. Despliega el menú de la derecha del paquete `acme-test-files_1.0-2_all.deb` y pulsa en Información del paquete.

Aquí podrás ver el registro de los cambios (entre otra información):

```
acme-test-files (1.0-2) unstable; urgency=low

 * Added file svg

-- Alberto Gacías <alberto@migasfree.org> Fri, 3 Feb 2018 18:25:00 +0100

acme-test-files (1.0-1) unstable; urgency=low

 * Ficheros de ejemplos

-- Alberto Gacías <alberto@migasfree.org> Sat, 18 May 2013 08:32:00 +0200
```

¿Qué equipos tienen el cambio `acme-test-files-1.0-2`?

Ve a Consultas - Ordenadores en producción con el paquete... Escribe en el campo Paquete `acme-test-files-1.0-2` y obtendrás el resultado.

2.2.5 Conclusión

Aunque el **empaquetado de los ECS** requiera un esfuerzo inicial, los beneficios que obtendrás justifican sobradamente el uso de este método:

- Dispondrás de sistemas más estables.
- Te permitirá hacer el seguimiento y control de los cambios.
- Y mejorarás la resolución de incidencias.

Beneficios de crear paquetes de configuración

- La configuración permacece encapsulada.
- Las configuraciones puede revertirse fácilmente.
- Facilita las pruebas antes del despliegue.
- Facilita la distribución de las configuraciones de forma segura.
- Proporciona integridad frente a los cambios de la configuración.

Desventajas del empaquetado de la configuración

- Cuesta más tiempo que otras alternativas ya que hay que crear los paquetes.

Beneficios de usar `migasfree`

Utilizar *migasfree* para la realizar la *Liberación* te permitirá:

- Controlar a quién y a partir de qué momento se deben aplicar los cambios.
- Tener una auditoría centralizada:
 - Inventario de Ordenadores.
 - Hardware.
 - Software (actual e histórico).
 - Inventario de los cambios.
 - y algunas cosas más, que te serán desveladas en los siguientes capítulos.

2.3 Configurando `migasfree-client`

La libertad no es poder elegir entre unas pocas opciones impuestas, sino tener el control de tu propia vida. La libertad no es elegir quien será tu amo, es no tener amo.

—Richard Stallman.

En el capítulo anterior nos hemos centrado en cómo se realiza el proceso de la GCS.

En este capítulo, vas a *configurar el cliente de migasfree* (mediante empaquetado) para que se conecte al servidor migasfree que ya debes tener funcionando.

Todos los comandos de este capítulo los vas a ejecutar en otra máquina virtual, con Ubuntu instalado, y que debes tener en la misma red en la que esté la máquina virtual del servidor.

El objetivo de este capítulo es que conozcas un poco más el empaquetado.

2.3.1 Instalando migasfree-client en Ubuntu

Instala el cliente *migasfree*:

```
# wget -O - http://migasfree.org/pub/install-client | bash
```

Observa como en el fichero `/etc/migasfree.conf` que ha instalado el paquete `migasfree-client` no hay, lógicamente, ningún ajuste configurado.

```
less /etc/migasfree.conf
```

A continuación, vamos a configurar este fichero haciendo uso del empaquetado, así que no lo hagas manualmente.

2.3.2 Obteniendo acme-migasfree-client

Al igual que hiciste con `acme-test-files`, puedes bajarte el fuente del paquete que vamos a utilizar de plantilla para configurar el cliente de migasfree.

En la nueva máquina virtual con Ubuntu, ejecuta el siguiente código si aún no te has descargado `fun-with-migasfree-examples`:

```
$ wget https://github.com/migasfree/fun-with-migasfree-examples/archive/master.zip
$ unzip master.zip
$ cd fun-with-migasfree-examples-master
```

2.3.3 Adaptando acme-migasfree-client

Accede al directorio `acme-migasfree-client` y observa su contenido:

```
$ cd acme-migasfree-client
$ ls -la
total 20
drwxrwxr-x 5 alberto alberto 4096 jun 18 20:54 .
drwxrwxr-x 4 alberto alberto 4096 jun 18 21:04 ..
drwxrwxr-x 3 alberto alberto 4096 jun 18 20:54 debian
drwxrwxr-x 3 alberto alberto 4096 jun 18 20:54 usr
```

Metadatos

Observa el directorio `debian`. Este directorio es el que contiene los metadatos del paquete. Los ficheros más importantes en este directorio son:

- El fichero `control` consiste en un conjunto de campos, representados en un formato común, que permiten al sistema de gestión de paquetes conocer los metadatos del paquete y así poder gestionarlo adecuadamente. Puedes consultar la [debian-policy](#) para explorar el conjunto de datos de `control`

- El fichero `changelog` contiene información, en un formato especial, con las modificaciones que se han realizado en cada versión del paquete. Cada vez que se modifica el paquete, hay que añadir una entrada en este fichero, incrementando la versión y registrando lo que se ha modificado.
- El fichero `copyright` contiene la información sobre los recursos, licencia y derechos de autoría de las fuentes originales del paquete.
- El fichero `rules` contiene las reglas que se utilizan para generar los paquetes a partir de sus fuentes.
- El fichero `install` contiene una lista de ficheros que serán instalados con el paquete.

Ahora que conoces el significado de estos ficheros, modifícalos cambiando el nombre del paquete `acme-migasfree-client` por `tuempresa-migasfree-client` y pon tu nombre y la fecha actual allí donde se requiera.

Modifica también el nombre del directorio raíz `acme-migasfree-client` por `tuempresa-migasfree-client`.

Scripts

Observa ahora los scripts `postinst` y `prerm`. Sus nombres nos indican cuando serán ejecutados por el sistema de gestión de paquetes.

- `postinst` inmediatamente después de que se produzca la instalación del paquete.
- `prerm` justo antes de que se produzca la eliminación del paquete.

Observa ahora el contenido de `postinst` y verás que aquí se hace una llamada al comando `dpkg-divert`. Mediante este comando hacemos lo que se conoce como una desviación de fichero (`divert`). Mediante la desviación, indicamos al sistema de gestión de paquetes que un fichero ya no pertenece a un determinado paquete sino al que nosotros establezcamos.

Así, el fichero de configuración `/etc/migasfree.conf`, que pertenece en principio al paquete `migasfree-client`, hacemos que pertenezca al paquete `tuempresa-migasfree-client` de tal manera que, una posible actualización de `migasfree-client` ya no nos afectará. Cada vez que queramos modificar un ajuste del cliente `migasfree` en `/etc/migasfree.conf`, lo haremos a través del fichero `usr/share/divert/etc/migasfree.conf` del paquete `tuempresa-migasfree-client`.

Fíjate también que en `prerm` deshacemos esta desviación, para que si desinstalamos el paquete, quede todo como estaba.

Modifica ahora el fichero `usr/share/divert/etc/migasfree.conf`. Tendrás que poner el ajuste `Server` con el nombre, o la IP, del servidor `migasfree` que hemos utilizado anteriormente, y el ajuste `Project` con el nombre de tu distribución, por ejemplo `ACME-1`. El resto de ajustes, modifícalos según tus intereses. Una vez hecho esto, y situado en el directorio `tuempresa-migasfree-client`, genera el paquete (debes tener el paquete `devscripts` y `debhelper` previamente instalados).

```
$ /usr/bin/debuild --no-tgz-check -us -uc
```

Con esto tendrás un paquete que configura el cliente `migasfree` para tu organización. Ahora, es momento de instalarlo:

```
# dpkg -i tuempresa-migasfree-client_1.0-1_all.deb
```

Observa que al instalar el paquete, `dpkg` te informa que se añade la desviación de `/etc/migasfree.conf`. Comprueba ahora que el ajuste `Server` y `Project` son los correctos.

```
# less /etc/migasfree.conf
```

Ahora ya estás preparado para registrar este ordenador en el servidor `migasfree`.

```
# migasfree -u
```

Comprueba que en el servidor se ha creado la versión ACME-1 y que existe un nuevo ordenador accediendo a la página web del servidor.

Finalmente, subimos el paquete a nuestro servidor migasfree con el fin de tenerlo disponible para su liberación a otros escritorios ACME-1.

```
# migasfree-upload -f tuempresa-migasfree-client_1.0-1_all.deb
```

- Introduce usuario: admin
- Contraseña: admin
- Proyecto: ACME-1
- Almacén: acme

2.3.4 Ejecución del cliente migasfree

Hasta ahora, siempre hemos ejecutado el cliente *migasfree* desde consola mediante el comando `migasfree -u` como `root`. Ahora vamos a hacer que se ejecute automáticamente cada vez que el usuario abra una sesión gráfica. Para este propósito, existe el paquete `migasfree-launcher`.

```
$ wget https://github.com/migasfree/migasfree-launcher/archive/latest.zip
$ unzip latest.zip
$ rm latest.zip
$ cd migasfree-launcher-latest
$ python setup.py --command-packages=stdeb.command bdist_deb
$ cd ..
```

Sube el fichero `migasfree-launcher` al servidor:

```
# migasfree-upload -f migasfree-launcher_1.0-1_all.deb
```

Ahora, observa los ficheros que contiene este paquete:

- `etc/sudoers.d/migasfree-launcher` establece los comandos que no requieren **password de root** para que pueden ser ejecutados desde un usuario cualquiera. Puedes obtener más información sobre la configuración de `sudoers` ejecutando `man sudoers` en un terminal.
- `etc/xdg/autostart/migasfree-indicator.desktop` ejecutará el comando `/usr/bin/migasfree-indicator` cuando el usuario inicia sesión gráfica. `migasfree-indicator` llamará a `/usr/bin/migasfree-launcher` y éste, a su vez, a `migasfree --update`.

Puedes aprender más sobre la especificación de los ficheros `.desktop` en freedesktop.org.

Ahora que ya tienes los paquetes `tuempesa-migasfree-client` y `migasfree-launcher` en el servidor *migasfree*, crea un despliegue en el servidor y pon estos paquetes en `paquetes` a instalar y asígnele el atributo `SET-ALL SYSTEMS`.

Nota: Para aprender más sobre el empaquetado, consulta la [Guía del nuevo desarrollador de Debian](#)

Nota: Para paquetería `rpm`, los metadatos del paquete se especifican en un único fichero llamado `SPEC`. Para aprender más sobre la creación de paquetes `rpm`, puedes consultar rpm.org y la [wiki del proyecto fedora](#).

2.3.5 Despliegue

A partir de este momento, vas a poder administrar fácilmente los escritorios Ubuntu de tu organización, de forma generalizada, instalando simplemente estos dos paquetes.

Hay varias formas de realizar esta instalación:

- Bajando los dos paquetes a cada uno de los escritorios e instalándolos mediante el comando `dpkg -i`
- Creando un fichero `/etc/apt/sources.list.d/migasfree.list` con el siguiente contenido:

```
deb http://<myserver>/public/<project>/REPOSITORIES <deployment> PKGS
```

donde sustituirás:

- `<myserver>` por tu servidor.
- `<project>` por el proyecto que pusiste en `/etc/migasfree.conf`
- y `<deployment>` por el nombre de un despliegue que tenga como paquetes disponibles: `tuempresa-migasfree-client`, `migasfree-client` y `migasfree-launcher`. Como paquetes a instalar puedes poner: `tuempresa-migasfree-client` y `migasfree-launcher`

Una vez creado este fichero, ejecuta:

```
# apt-get update
# apt-get install tuempresa-migasfree-client
# migasfree -u
```

y tendrás instalados los 3 paquetes.

- Puedes hacer un clon de un equipo donde ya estén instalados estos paquetes, utilizando un sistema de clonado como [clonezilla](#). Este es el método que usamos en **AZLinux**, y nos resulta muy cómodo y rápido ya que en una memoria USB llevamos un *clonezilla*, junto con la imagen clonada de nuestro escritorio, consiguiendo instalar un AZLinux en menos de 10 minutos.
- Puedes crear un DVD de tu escritorio tal y como se realiza en el proyecto [vitalinux](#). En concreto, tendrías que adaptar el paquete `vx-create-iso` a tus necesidades. En este método son los usuarios quienes se bajan la ISO del DVD y se instalan ellos mismos el sistema.

La reacción al cacahuete.

3.1 La configuración del sistema migasfree

El hombre razonable se adapta al mundo; el irrazonable intenta adaptar el mundo a sí mismo.
Así pues, el progreso depende del irrazonable.

—George Bernard Shaw

En los capítulos anteriores, has aprendido a instalar el servidor y el cliente *migasfree*, así como a crear paquetes. La creación de paquetes no es una tarea trivial, no tanto por su construcción en sí, sino por el hecho de que son necesarios amplios conocimientos de los sistemas operativos y de las aplicaciones.

En este y en los siguientes tres capítulos, vas a aprender a adaptar y usar el servidor *migasfree*.

3.1.1 Fórmulas

En *migasfree*, una *fórmula* es una característica de los equipos o de los usuarios, y que nos servirá para desplegar los paquetes.

Como administrador de *migasfree*, una de las primeras tareas que debes realizar es definir estas fórmulas. Debes preguntarte en función de qué características vas a realizar los despliegues. Por ejemplo, ¿te interesa desplegar los paquetes por el HOSTNAME de los equipos?, ¿y por subred? ¿Qué tal por el grupo al que pertenece el usuario en el LDAP? ¿O por su contexto LDAP?

Nota: En AZLinux, usamos principalmente el contexto LDAP al que pertenece el usuario para desplegar los cambios por los distintos servicios o departamentos de nuestro ayuntamiento, y en menor medida usamos también el CID.

Una *fórmula* es un código que se programa en un registro de la base de datos de *migasfree*. Estas fórmulas serán ejecutadas en cada uno de los clientes *migasfree* y sus valores de retorno serán devueltos al servidor como *atributos*.

Nota: El atributo es el valor concreto que toma una fórmula al ser ejecutada en un equipo.

Veamos un ejemplo sencillo de todo esto con la fórmula *MACHINE NAME*. Accede a la web de tu servidor *migas-free* y ve a Configuración-Fórmulas-MACHINE NAME. En este registro, verás el siguiente código escrito en python.

```
import platform
print platform.node()
```

Si ejecutas `python` en una consola y escribes estas dos líneas, verás que *python* muestra, por la salida estándar, el nombre de tu equipo.

```
$ python
Python 2.7.3 (default, Apr 10 2013, 05:46:21)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import platform
>>> print platform.node()
white
```

En mi caso me ha devuelto `white`, que es el nombre de mi portátil. ¿A que no aciertas de qué color es?

Esto es, en definitiva, lo que hace el cliente *migasfree*: obtiene del servidor la fórmula *MACHINE NAME* (su código), la ejecuta y devuelve al servidor el resultado como atributo (`HST-white`).

Podrías haber escrito la fórmula también en código `bash` simplemente llamando al comando de Linux `hostname` o escribiendo `echo $HOSTNAME` (el resultado sería el mismo), pero utilizar código `python` nos permite, en este caso, usar la misma fórmula también para plataformas Windows o Mac OS.

Campos de la Fórmula

Observa cada uno de los campos de la *fórmula*:

- **Prefijo:** Es una combinación de tres números o letras. Este prefijo se utiliza para agrupar e identificar los atributos.
- **Nombre:** Denomina a la fórmula.
- **Habilitado:** Indica si está activa la fórmula. Si no está marcado, la fórmula no será ejecutada en los clientes.
- **Lenguaje de programación:** En el que está escrito el código de la fórmula.
- **Código:** Instrucciones a ejecutar en los clientes para obtener atributos.
- **Clase:** Hay cuatro tipos de clases y que nos permiten tratar el valor devuelto por la fórmula de diferentes maneras:

- **Normal.** El valor devuelto por la fórmula viene con el siguiente formato:

```
<valor>~<Descripción>
```

o, simplemente, como:

```
<valor>
```

- **Lista:** El valor al ejecutar la fórmula en el cliente es una lista de atributos separados por una coma. Puedes ver un ejemplo en la fórmula *PCI*. Su formato es:

```
<valor>~<Descripción>, ...
```

- **Añadir por la derecha:** Permite añadir atributos de la siguiente manera: Si el valor devuelto por la fórmula es «CONTEXTO1.CONTEXTO2.MIEMPRESA», el servidor interpreta que el equipo tiene estos tres atributos:

- MIEMPRESA
- CONTEXTO2.MIEMPRESA
- CONTEXTO1.CONTEXTO2.MIEMPRESA

Se utiliza esta clase para crear atributos relacionados con LDAP.

- **Añadir por la izquierda.** Lo mismo que la anterior, pero agregando por la izquierda.

- CONTEXTO1
- CONTEXTO1.CONTEXTO2
- CONTEXTO1.CONTEXTO2.MIEMPRESA

- **Añadir automáticamente:** Si este campo está marcado, los nuevos atributos serán añadidos automáticamente a la base de datos de migasfree. En caso contrario es el administrador de migasfree el responsable de añadir manualmente los atributos para esta fórmula. Se pueden añadir atributos manualmente accediendo a Datos-Atributos.

Fórmulas específicas

Existen unas fórmulas predefinidas que tienen unos objetivos muy concretos y que no pueden ser eliminadas del sistema. Lo más característico de ellas es que **no son ejecutadas en el cliente** sino en el servidor.

- **SET:** Esta fórmula tiene un atributo llamado SET-ALL SYSTEMS. Todos los ordenadores tendrán este atributo sin excepción. Sirve para referirse a **todos** los ordenadores. Por ejemplo, si en un despliegue asignas este atributo, todos los ordenadores tendrán acceso a él. Es habitual usarlo también en la última demora de un calendario. Esta fórmula, además, se usa internamente para definir *Conjuntos de Atributos*.
- **CID:** *Computer Identifier*. Esta fórmula generará un atributo que es igual al campo `id` de la tabla `computer` de la Base de Datos de migasfree.

Dicho atributo CID es único por cada ordenador y se utiliza en lugar de referirse al UUID de la placa base de un ordenador siendo **el más recomendado** para señalar a un ordenador en concreto (procesos como el *Reemplazo de ordenadores* tienen en cuenta este atributo).

Por ejemplo, un atributo CID-572 se correspondería con el UUID 5FD85780-9BC9-11E3-91B8-F0921CF3678D.

El ajuste *MIGASFREE_COMPUTER_SEARCH_FIELDS* del servidor, permite configurar búsquedas por otros campos del ordenador a la hora de asignar un determinado CID en el servidor migasfree.

Cuando un ordenador pasa a un estado Baja o a Disponible todos sus CID asignados en el sistema son eliminados. Ver campo estado de *Ordenadores*.

El CID aparece por defecto en la etiqueta del ordenador que muestra el comando `migasfree-label`.

3.1.2 Categorías de etiquetas

Hasta ahora, has visto que una fórmula es un código que se ejecuta en el cliente para obtener un atributo automáticamente. Ahora bien, pueden existir casos en que no se puede obtener automáticamente

estos atributos. Imagina que quieres «etiquetar» ciertos equipos según la funcionalidad que van a realizar (tratamiento gráfico, administración, aula, etc.). Esto no es algo que, a priori, se pueda programar.

En migasfree existe la posibilidad de crear estas etiquetas y asignarlas manualmente a los equipos tal y como harías con una etiqueta física que pegas a un ordenador.

Una etiqueta no ejecutará ningún código en el cliente. Es el propio registro del ordenador en el servidor de migasfree quien lleva asignadas, manualmente, estas etiquetas. A todos los efectos, una etiqueta es un atributo más del sistema y, por tanto, te permitirá hacer el despliegue también en función de ellas.

Por cada Configuración-Categorías de etiquetas, existirá un conjunto de etiquetas que manualmente debes añadir en Datos-Etiquetas. Una vez añadidas, puedes asignarlas a Datos-Ordenadores. También puedes editar Datos-Etiquetas y asignarle un conjunto de ordenadores.

Existe en el cliente el comando `migasfree-tags` que permite consultar y asignar etiquetas desde el propio cliente.

Para obtener el conjunto de etiquetas que pueden ser asignadas a un ordenador ejecuta:

```
migasfree-tags --available
```

Para consultar las etiquetas asignadas a un ordenador ejecuta:

```
migasfree-tags --get
```

Para asignar etiquetas al equipo, seleccionando manualmente las etiquetas entre las disponibles en el sistema, ejecuta:

```
migasfree-tags --set
```

Para asignar determinadas etiquetas a un equipo, escribe las etiquetas separadas por espacios:

```
migasfree-tags --set <ETIQUETA1> <ETIQUETA2> ...
```

Para asignar etiquetas en el servidor migasfree pero que no se produzca **ningún cambio de paquetes** utiliza:

```
migasfree-tags --communicate <ETIQUETA1> <ETIQUETA2> ...
```

Para quitar todas las etiquetas de un equipo, ejecuta:

```
migasfree-tags --set ""
```

Las etiquetas están relacionadas con los campos de los despliegues:

- Paquetes pre-incluidos por defecto
- Paquetes incluidos por defecto
- Paquetes excluidos por defecto

ya que al ejecutar el comando `migasfree-tags --set` se instalarán los paquetes definidos en los `pre-incluidos` e `incluidos` y se desinstalarán los paquetes definidos en el campo `excluidos`, siempre y cuando los atributos asignados al repositorio coincidan con los del equipo. Esto se utiliza para crear la imagen ISO de los escritorios.

Nota: En AZLinux, usamos `migasfree-tags` básicamente para, partiendo de una imagen ISO de Ubuntu, desinstalar e instalar los paquetes que componen nuestro escritorio y crear una imagen del disco para clonar.

Nota: En Vitalinux, se emplean las etiquetas para cambiar fácilmente de «sabor». Cuando se quiere cambiar de «sabor» (Infantil, Primaria, Profes, ...), simplemente se eligen las etiquetas mediante el comando `migasfree-tag --set`, produciéndose automáticamente la instalación y desinstalación de los paquetes correspondientes. También se utiliza en la creación del DVD, permitiendo hacer una ISO para cada sabor o conjunto de sabores.

Campos de categorías de etiquetas

- **Prefijo:** Es una combinación de tres números o letras. Este prefijo se utiliza para agrupar e identificar las etiquetas.
- **Nombre:** Denomina el tipo de etiqueta.
- **Habilitado:** Si no está marcado, las etiquetas de este tipo no serán funcionales.
- **Clase:** El funcionamiento es exactamente igual al campo de mismo nombre que tienen las fórmulas.

Un valor muy útil que puede tomar este campo es el de `añadir por la derecha`. Imagina que quieres agrupar los ordenadores por ubicación para liberar software por distintas zonas. Una forma de hacerlo es crear una Categoría de etiqueta llamada p.e. UBICACIÓN definida de clase `añadir por la derecha`. Después, puedes crear las Etiquetas de tipo UBICACION p.e.:

```
UBI-PLANTA-1.SEDE_CENTRAL.MADRID
```

Cuando un equipo con esta etiqueta asignada se conecta al servidor, automáticamente el servidor interpretará que tiene no una, sino tres etiquetas:

```
UBI-MADRID
UBI-SEDE_CENTRAL.MADRID
UBI-PLANTA-1.SEDE_CENTRAL.MADRID
```

Con lo que finalmente podemos liberar software a todo MADRID, a toda la sede central de Madrid, o solamente a la planta 1ª.

Nota: Observa que el caracter de delimitación es el punto: .

3.1.3 Conjuntos de Atributos

En ocasiones puedes necesitar agrupar Atributos.

Imagina que tienes muchos equipos a los que asignar una cierta Etiqueta y que te resulta pesado tener que hacerlo uno a uno. Puedes entonces crear un Conjunto de Atributos.

Supón que tienes subredes con un buen ancho de banda y otras subredes que no, y que necesitas liberar software en función de esto. Podríamos crear dos Conjuntos de Atributos:

```
Conjunto 1:
Nombre:          RED LENTA
Atributos asignados: NET-192.168.1.0/24
                  NET-192.168.8.0/24

Conjunto 2:
Nombre:          RED RAPIDA
Atributos asignados: SET-ALL SYSTEMS
Atributos excluidos: SET-RED LENTA
```

De esta manera, cualquier equipo de las subredes 192.168.1.0/24 o 192.168.8.0/24, al ejecutar `migasfree -u`, se le asignará automáticamente un `Atributo: SET-RED LENTA`. Al resto de equipos se le asignará el `Atributo: SET-RED RAPIDA`.

Ahora ya podríamos crear `Despliegues` y asignarles dichos `Atributos`.

Los `Conjuntos de Atributos` no ejecutan ningún código en el cliente, sino que son evaluados en el servidor. Si un ordenador pertenece a un conjunto, se le asigna un `Atributo` con el mismo nombre que el `Conjunto de Atributos`.

Campos de Conjuntos de Atributos

- **Nombre:** Denomina al conjunto.
- **Habilitado:** Indica si el conjunto será evaluado.
- **Atributos incluidos:** Lista de `Atributos` que formarán parte el conjunto.
- **Atributos excluidos:** Lista de `Atributos` a excluir del conjunto.

3.1.4 Proyectos

Migasfree puede trabajar con distintos `Sistemas Operativos`. Un `proyecto`, en migasfree, representa a un conjunto de ordenadores que comparten un mismo `Sistema Operativo base`.

Por ejemplo, en el `Ayto. de Zaragoza` contamos con los siguientes `proyectos`:

- AZLinux-1 (SLED 10.2) Se migró a AZLinux-2 (ningún PC).
- AZLinux-2 (OpenSUSE 11.2) Se migró a AZLinux-12 (ningún PC).
- AZLinux-12 (Ubuntu 12.04) En producción (900 PC). En fase de migración a otros AZLinux.
- AZLinux-14 (Ubuntu 14.04) En producción (500 PC).
- AZLinux-16 (Ubuntu 16.04). En producción (100 PC).
- ZA (Ubuntu 10.04 para escritorios tipo kioskos). Obsoleto.
- WIN-XP (Windows XP). En producción (1600 PC).
- AZW-10 (Windows 10). En producción (80 PC).

Cada ordenador estará configurado en un único `proyecto` en un momento dado. Cambios de `proyecto` en un ordenador crean en el sistema un registro de `migración` automáticamente. De esta manera, es posible conocer las diferentes migraciones de `S.O.` que se han ido produciendo en los equipos y en qué momento se han hecho efectivas. Puedes consultar las migraciones accediendo a `Datos-Migraciones`.

Mediante el ajuste `MIGASFREE_AUTOREGISTER` se permite, o no, a los equipos registrar automáticamente los `proyectos`. Puedes consultarlo en *[Ajustes del servidor migasfree](#)*.

Campos del proyecto

- **Nombre:** Denomina al proyecto.
- **Plataforma:** a la que pertenece el proyecto.
- **Sistema de gestión de paquetes:** El `PMS` que se utiliza en el sistema operativo de este proyecto.

- **Auto registrar ordenadores:** Si está marcado, se permiten registrar ordenadores desde un cliente automáticamente. En este caso, sólo con que un equipo esté configurado con el proyecto, será añadido automáticamente a la base datos.

En caso contrario, sólo se podrán registrar ordenadores mediante el uso de un usuario que cuente con los permisos adecuados para añadir ordenadores al sistema.

3.1.5 Plataformas

Los proyectos se clasifican por plataformas. Las plataformas vienen establecidas por la función `python platform.system()` y, por tanto, sus valores pueden ser:

- Linux
- Windows
- (Otras)

Esta clasificación de los proyectos te permite realizar consultas y estadísticas en función de la plataforma.

Mediante el ajuste `MIGASFREE_AUTOREGISTER` se permite, o no, a los equipos registrar automáticamente las plataformas. Puedes consultarlo en [Ajustes del servidor migasfree](#).

3.1.6 Perfiles de usuario Migasfree

En *migasfree* existen dos tipos de usuarios, los usuarios que administran *migasfree* y los usuarios que utilizan los ordenadores. Este apartado se refiere a los primeros.

Cuando se genera la base de datos de *migasfree*, se crean 7 grupos de usuarios y 8 usuarios predeterminados:

Grupos de Usuarios

En función de las tareas que los usuarios de administración de migasfree pueden realizar, se establecen los siguientes grupos de usuarios.

- `Configurator` con permisos de lectura/escritura a:
 - Fórmulas
 - Proyectos
 - Sistemas de gestión de paquetes (PMS)
 - Plataformas
 - Comprobaciones
 - Definiciones de fallas
 - Mensajes
 - Sincronizaciones
 - Mensajes del servidor
 - Migraciones
 - Notificaciones
- `Computer Checker` tiene permisos de lectura/escritura a:
 - Errores

- Fallas
- Mensajes
- Sincronizaciones
- `Liberator`. Permisos de lectura/escritura a:
 - Despliegues
 - Calendarios
- `Packager` cuenta con permisos de lectura/escritura a:
 - Paquetes
 - Almacenes
- `Query`. Permisos de lectura/escritura a:
 - Consultas
- `Device installer` cuenta con permisos de lectura/escritura a:
 - Dispositivos
- `Reader`. Permisos de sólo lectura a todas las tablas.

Usuarios

- `admin`. Tiene permisos de lectura/escritura a todas las tablas.
- `packager`. Pertenece a los grupos `Reader` y `Packager`.
- `configurator`. Pertenece a los grupos `Reader` y `Configurator`.
- `installer`. Pertenece a los grupos `Reader` y `Device installer`.
- `query`. Pertenece a los grupos `Reader` y `Query`.
- `liberator`. Pertenece a los grupos `Reader` y `Liberator`.
- `checker`. Pertenece a los grupos `Reader` y `Computer Checker`.
- `reader`. Pertenece al grupo `Reader`.

Estos usuarios tienen por defecto como contraseña su nombre, es decir, la contraseña de `admin` es `admin`, y lo mismo es aplicable al resto de usuarios.

Estos usuarios son ficticios para realizar pruebas y conviene que sean eliminados. Se recomienda crear los usuarios reales que usarán el servidor *migasfree*, asignándoles los grupos de usuarios correspondientes.

Nota: Es importante que en un entorno de producción se deshabiliten los usuarios que no se vayan a utilizar o que, al menos, se les cambie la contraseña por motivos de seguridad.

Cambio de contraseña

La contraseña puede ser cambiada por los usuarios pulsando en su nombre de usuario y que aparece arriba a la derecha en todas las páginas web de la aplicación.

También puede ser modificada por otro usuario que tenga marcado el campo `Es superusuario`, accediendo al registro del usuario en cuestión y modificando directamente su campo `Contraseña`.

Proyecto por defecto de un usuario

Los usuarios tienen un campo `proyecto` que sirve para filtrar registros. De esta manera, cuando un usuario consulta los despliegues p.e., se muestran por defecto los despliegues del proyecto que tiene asignado. Este filtro por defecto se aplica a despliegues, almacenes y paquetes.

Un usuario puede seleccionar su proyecto pulsando en su nombre de usuario y luego `Preferencias`

3.1.7 Comprobaciones

Son un conjunto de comprobaciones que se realizan para alertar al usuario. Pulsando en cada una de las `Alertas` puedes obtener más información. Ver figura 8.1.

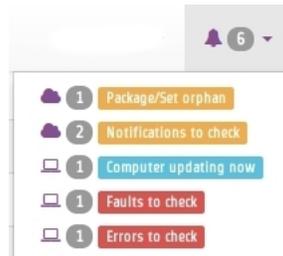


Figura 1: Alertas del sistema.

Hay 9 comprobaciones predeterminadas en *migasfree*:

- `Errores sin comprobar`. Cuando en un cliente *migasfree* se produce algún error, es enviado al servidor. Esta comprobación hace que se muestren estos errores. Una vez revisado o solucionado un error en el cliente debes editar el error en el servidor y marcar el campo `comprobado`. Esto hará que ya no aparezca en la lista de errores a comprobar. Puedes también seleccionar un conjunto de errores en la lista de errores y en el desplegable de acción seleccionar `La comprobación es correcta`.
- `Fallas sin comprobar`. Cuando en un cliente *migasfree* se produce una falla, es enviada al servidor. Esta comprobación hace que se muestren las fallas pendientes. La manera de proceder con las fallas es similar a la de los `Errores sin comprobar`.
- `Notificaciones sin comprobar`. Son hechos que se han producido en el sistema y que son informados mediante esta comprobación. Un ejemplo de notificación es cuando un equipo da de alta una plataforma o un proyecto nuevo en el sistema.
- `Paquetes huérfanos`. Comprueba si hay paquetes que no están asignados a ningún despliegue.
- `Ordenadores sincronizándose`. Cuando un equipo está ejecutando el cliente *migasfree*, éste va informando al servidor de lo que está haciendo mediante un texto que indica el proceso que está realizando. Cuando el cliente *migasfree* finaliza, envía al servidor un mensaje de texto vacío. Esta comprobación comprueba cuántos de estos mensajes se han recibido.
- `Ordenadores retrasados`. Si pasa un determinado tiempo desde que se recibió el último mensaje del cliente, es muy posible que algo ha ido mal en el cliente. Quizás perdió la conexión, o el usuario apagó el equipo en medio de la ejecución del cliente *migasfree*, o quizás ha habido algún error. Esta comprobación permite detectar estos casos. La cantidad de tiempo viene establecida por defecto en 30 minutos y puede ser modificado mediante el ajuste `MIGASFREE_SECONDS_MESSAGE_ALERT` de los *Ajustes del servidor migasfree*.
- `Generación de repositorios`. Indica si se están generando los metadatos de algún repositorio físico asociado a algún despliegue.
- `Despliegues con calendario activo`.

- **Despliegues con calendario finalizado.** Si hay despliegues que tienen el calendario de distribución finalizado, deberías pasar esa información (paquetes disponibles, a instalar, etc.) a otro despliegue que no tenga calendario y borrar el despliegue original para simplificar la gestión de los mismos.

Las alertas proporcionan al usuario una vista general de la situación actual del sistema, dirigiendo su actuación a lo relevante.

El objetivo en todo momento debería ser mantener el sistema con 0 alertas. Esto indicaría que se han revisado los errores, se han comprobado las fallas, no hay paquetes huérfanos, etc.

3.1.8 Fallas

Una falla es un hecho negativo que se produce en un equipo cliente. Por ejemplo que un equipo se quede con poco espacio en la partición de sistema, es algo a lo que se debe prestar atención y ser solucionado antes de que sea tarde.

Migasfree, mediante las fallas, permite lanzar código en el cliente con este objetivo. Fíjate que las posibilidades son inmensas y que te permite ser muy proactivo.

En definitiva, una falla es un código que se ejecuta en el cliente. Si el código escribe algo por la salida estándar, ésta será enviada al servidor como `Falla`. El servidor entonces añadirá un registro de `Falla`, apareciendo en las `Alertas` de los usuarios de `migasfree`.

Campos de Definición de Falla

- **Nombre:** Denomina a la falla.
- **Habilitado:** Activa o desactiva la falla.
- **Descripción:** Para detallar lo que hace la falla.
- **Lenguaje de programación:** Especifica en qué lenguaje está escrito el código. Mi recomendación es que programes en la medida de lo posible en **python**.
- **Código:** Instrucciones que detectan alguna falla en los equipos y que debe poner en la salida estándar un texto que indique la falla producida. Puede ser útil en algunos casos poner también el procedimiento a seguir.
- **Atributos incluidos:** Permite asignar en qué equipos cliente será efectiva la falla. Por ejemplo si escribes el código en bash, deberías asignar la falla sólo a los equipos con plataforma Linux `PLT-Linux`, ya que plataformas Windows no serán capaces de ejecutar bash. También te puede interesar programar una falla sólo para obtener información de un equipo o de un grupo de equipos.
- **Atributos excluidos:** Permite excluir a ciertos equipos de la ejecución de la falla.
- **Usuarios:** Sirve para asignar usuarios de `migasfree` a los que les aparecerán las fallas de este tipo cuando se accede desde las `Alertas` (sólo se muestran las que están pendientes de comprobar por el usuario autenticado).

Si una definición de falla no tiene asignado ningún usuario, las fallas que se produzcan aparecerán a cualquier usuario autenticado.

Nota: Poder ejecutar código en los clientes proporciona una gran potencia para realizar cualquier cosa. Usa esta capacidad con responsabilidad y sé meticuloso en las comprobaciones antes de activar cualquier falla.

3.1.9 Consultas

Migasfree incorpora un sistema para crear consultas parametrizables.

Cada consulta se programa en un registro y podrá ser ejecutada accediendo a `Consultas`.

Hay una pocas consultas ya predefinidas, pero puedes programar nuevas o adaptar las que ya existen.

Campos de consulta

- **Nombre:** Denomina la consulta.
- **Descripción:** Describe la consulta.
- **Código:** Instrucción en *Django* de la consulta. Mediante la asignación de unas variables predeterminadas el servidor podrá crear la consulta.

Las variables en concreto son:

- **query:** Conjunto de registros de la consulta.
- **fields:** Lista de los campos del *QuerySet* que se quieren mostrar.
- **titles:** Lista de los títulos de los campos que se quieren mostrar.
- **project:** Sirve para obtener el proyecto del usuario y poder hacer filtros cuando se requiera.
- **Parámetros:** Permite la petición de parámetros de consulta. Se debe crear una función que se llame `form_params` y que devuelva una clase que herede de `ParametersForm`.

Nota: Para realizar consultas, necesitarás conocer un poco los `QuerySet` de Django y la Documentación del modelo de datos. Esta última la tienes disponible al final de todas las páginas de la aplicación, pulsando sobre el icono de base de datos.

3.1.10 Errores autocomprobables

Por defecto, los errores producidos por el PMS, se añaden al sistema como no comprobados. Ahora bien, en ocasiones puede resultar tedioso tener que marcar como comprobados uno a uno ciertos errores que, más que errores, son «alertas».

Para automatizar esta tarea puedes crear un `error` autocomprobable. Simplemente añade un registro con el `patrón de búsqueda` deseado y los errores que coincidan con ese patrón se marcarán automáticamente como comprobados.

Por ejemplo, si quisieras que todos los errores que llegan del tipo:

```
2014-10-03 10:44:47
Error: Generic error
Info: Curl error: Couldn't resolve host 'myserver'
```

se autocomprobaran, podrías emplear el siguiente patrón:

```
.*\sError: Generic error\sInfo: Curl error: Couldn't resolve host 'myserver'
```

3.2 La Liberación

El conocimiento nos hace responsables.

—Ernesto Guevara.

Este es el capítulo que mejor define a *migasfree*, ya que su principal funcionalidad es la de ofrecer unos determinados repositorios de paquetes que estarán disponibles para los clientes en función de sus atributos.

En los proyectos de software libre, la liberación tiene que ver con poner a disposición de la comunidad un determinado software. Aspectos como la autoría o la licencia son esenciales, tanto o más como el propio software que se libera.

Liberar software en *migasfree* implica, además, decidir a quién y a partir de qué momento, un cliente tendrá acceso a dicho software.

3.2.1 Subiendo paquetes al servidor

Antes de poder liberar el software, obviamente, tienes que subirlo al servidor.

Como viste en los primeros capítulos, la manera de hacerlo es utilizando el comando de cliente:

```
migasfree-upload -f <mipaquete>
```

o si quieres subir un conjunto de paquetes (Set) ponlos todos juntos en un directorio y ejecuta:

```
migasfree-upload -d <midirectorio>
```

Para subir paquetes al servidor es necesario utilizar un usuario que tenga permisos de lectura/escritura en la tabla de almacenes y paquetes. Por defecto el usuario `packager` y el usuario `admin` los tienen.

Para no tener que introducir cada vez que subas un paquete al servidor el usuario, su contraseña y/o el proyecto con el que trabajas, puedes asignar los ajustes indicados en la sección [Packager] de *Ajustes del cliente migasfree*.

3.2.2 Almacenes

Un almacén es una ubicación o ruta del servidor donde se colocan los paquetes y/o conjuntos de paquetes subidos al servidor. No es más que un directorio colgando de la ruta `/var/migasfree/repo/<PROJECT>/STORES`, y que se utiliza para tener organizados los paquetes. También es accesible desde un explorador web accediendo a la ruta:

```
http://tuservidor/public/<PROJECT>/STORES/
```

Lo anteriormente expuesto, corresponde al lugar donde se almacenan los archivos del paquete, pero además hay una parte lógica que es necesaria llevar en la base de datos de *migasfree*. Es lo que denominamos registros de «Almacén».

Cuando se utiliza el comando `migasfree-upload` y se indica una ubicación inexistente, el servidor automáticamente creará el registro lógico en la base de datos y creará la carpeta en el sistema de archivos.

Campos de Almacén

- **Nombre:** Denomina al almacén. Corresponde al nombre de la carpeta en el sistema de archivos.
- **Proyecto.** Indica el proyecto *migasfree* al que pertenece el almacén.

3.2.3 Paquetes

Cuando subes un paquete o un conjunto de paquetes al servidor, además de copiarse en el almacén o ubicación indicada, se crea un registro lógico en la base de datos. Estos registros nos servirán para asignarlos posteriormente en los Despliegues que vayamos creando.

Campos de Paquete

- **Nombre:** Es el nombre del fichero del paquete.
- **Proyecto:** Indica el proyecto *migasfree* al que pertenece el paquete.
- **Almacén:** Especifica la ubicación donde está situado el paquete.

Acciones de Paquete

A la derecha del nombre del paquete, en la lista de paquetes, hay un desplegable con las siguientes acciones:

- **Información del paquete.** Permite ver los metadatos del paquete.
- **Descargar.** Permite almacenar el paquete seleccionado en tu equipo.

Si necesitas borrar uno o varios paquetes, selecciónalos y en el desplegable Acción elige Eliminar Paquetes/conjuntos seleccionados y después pulsa en el botón ir.

- **Eliminar Paquetes/conjuntos seleccionados.** Permite borrar el registro del paquete. A medida que vayas haciendo cambios en el software, irás teniendo distintas versiones del mismo paquete. Generalmente, te interesará trabajar sólo con la última versión. Si quieres que sólo te aparezca ésta a la hora de asignarlo a los Despliegues, puedes borrar los registros de Paquetes antiguos. Borrar el registro no borrará el archivo del paquete en ningún caso y simplificarás la selección de paquetes.

Paquetes huérfanos

Un paquete huérfano es un paquete que no está asignado a ningún despliegue. Cuando un paquete es subido al servidor, o cuando lo quitas de un despliegue y no está en ningún otro, se convierte en un paquete huérfano. Existe una comprobación de Alerta que te avisará de cuáles son estos paquetes.

3.2.4 Información de los paquetes

Si accedes a Liberación-Información de paquetes verás que te aparecen dos carpetas:

- **STORES.** Muestra esta carpeta, en donde podrás navegar hasta un determinado paquete que hayas subido previamente.
- **REPOSITORIES** Muestra los repositorios físicos (en el sistema de archivos) que se hayan creado, y que son los que, en última instancia, verán los clientes. En realidad, los paquetes que veas en REPOSITORIES no son más que enlaces simbólicos a los paquetes ubicados en STORES.

Si quieres ver los metadatos de un determinado paquete, simplemente, pulsa sobre él.

3.2.5 Despliegues

Me gusta la definición: **migasfree es simplemente un gestor de despliegues de paquetes**. En realidad es básicamente esto. De hecho, así es como empezó este proyecto, y a partir de aquí ha ido creciendo hasta convertirse en lo que es hoy en día, un gestor de sistemas.

A todos los efectos, y desde el punto de vista del cliente, un despliegue en *migasfree* es un repositorio de paquetes estándar como los que puedes encontrar en cualquier distribución. *Migasfree* permite crear muy fácilmente estos repositorios y asignarlos a los equipos en función de sus atributos a partir de una fecha determinada.

Campos de despliegue

- **Nombre:** Denomina al despliegue.

Nota: En AZLinux solemos incorporar en el nombre del despliegue el número de tarea de redmine al que hace referencia el cambio de software que queremos liberar.

- **Proyecto:** Especifica el proyecto en el que estará disponible el despliegue.
- **Habilitado:** Activa o desactiva el despliegue.
- **Comentario:** Campo de texto que sirve para registrar aclaraciones sobre el despliegue. Es muy conveniente que registres las modificaciones que vayas haciendo en este campo, indicando quién, cuándo y qué se ha modificado.

Un ejemplo de cómo lo hacemos en AZLinux, sería:

```
[alberto@2013-03-09] Añadido paquete azl-firefox-12.0-3_all.deb
[alberto@2013-04-10] Añadido paquete azl-firefox-12.0-4_all.deb
[eduardo@2013-05-10] Detectado problemas en algunos clientes. Desactivo
el despliegue hasta diagnosticar y encontrar solución.
```

- El qué (paquetes):
 - **Paquetes disponibles:** En este campo se seleccionan los paquetes y/o conjuntos de paquetes que se incluirán en el repositorio físico.

Que un paquete esté incluido en un repositorio y el repositorio sea accesible desde el cliente, no implica que se instale el paquete. Los sistemas de paquetería sólo actualizan aquellos paquetes que ya estuvieran instalados en el sistema.

Cada vez que hay una modificación de este campo y se pulsa el botón *Grabar*, se generarán los metadatos del repositorio físico. Dependiendo de la cantidad de paquetes que se tengan que procesar, el tiempo para realizar este proceso puede ser largo. En los casos en los que se asigne un conjunto de paquetes donde se incluyan todos los paquetes de un DVD p.e. puede llegar a ser del orden de decenas de minutos.

Nota: Fíjate que aparecen sólo los paquetes (los subidos individualmente) más los conjuntos de paquetes a la hora de seleccionarlos en los despliegues. Los paquetes incluidos dentro de los conjuntos de paquetes no pueden asignarse individualmente. Esto es así para simplificar y hacer más sencilla la asignación de paquetes y no perdernos entre los miles que componen una distribución.

- **Paquetes a instalar:** Campo de texto que especifica una lista de paquetes separados por espacios o por retornos de carro. Estos paquetes serán instalados **obligatoriamente** a los clientes que tengan acceso al despliegue.

Se puede especificar sólo el nombre del paquete, o el nombre de paquete más una versión.

Este campo se tiene en cuenta al ejecutar los comandos de cliente `migasfree --update` y `migasfree-tags --set`.

- **Paquetes a desinstalar:** Campo de texto que especifica una lista de paquetes separados por espacios o por retornos de carro que serán desinstalados **obligatoriamente** en los clientes.

Este campo se tiene en cuenta al ejecutar los comandos de cliente `migasfree --update` y `migasfree-tags --set`.

- **Por defecto:**

- **Paquetes pre-incluidos por defecto:** Campo de texto que especifica una lista de paquetes separados por espacios o por retornos de carro. Este campo sirve para instalar paquetes que configuran repositorios externos a migasfree. Un ejemplo de este tipo de paquetes lo tienes en el paquete `vx-repo-unizar`.

La razón de la existencia de este campo, es que después de instalar el repositorio externo, es necesario obtener de nuevo los metadatos de los repositorios (`apt-get update`), a fin de que el cliente tenga acceso inmediatamente a los paquetes contenidos en el repositorio externo.

Estos paquetes serán instalados a los clientes que tengan acceso al despliegue al ejecutar el comando `migasfree-tags --set`.

- **Paquetes incluidos por defecto:** Campo de texto que especifica una lista de paquetes separados por espacios o por retornos de carro. Estos paquetes serán instalados a los clientes que tengan acceso al despliegue al ejecutar el comando `migasfree-tags --set`.
- **Paquetes excluidos por defecto:** Campo de texto que especifica una lista de paquetes separados por espacios o por retornos de carro que serán desinstalados en los clientes que tengan acceso al despliegue al ejecutar el comando `migasfree-tags --set`.

- **A quién (atributos):**

- **Atributos incluidos:** Aquellos clientes que tengan un atributo que coincida con los asignados en este campo tendrán accesible el despliegue (a menos que otro atributo lo excluya).
- **Atributos excluidos:** Sirve para excluir atributos de la lista anterior.

Por ejemplo, si quieres liberar un paquete a toda la subred `192.168.92.0` menos al equipo `PC13098`, puedes hacerlo asignando:

- Atributos incluidos: `NET-192.168.92.0/24`
- Atributos excluidos: `HST-PC13098`

- **Cuándo (calendario):**

- **Fecha de inicio:** A partir de la cual estará disponible el despliegue en los clientes.
- **Calendario:** Especifica una programación del despliegue basada en calendario. En el siguiente apartado tienes más información.

3.2.6 Calendarios

Los calendarios te permiten programar sistemáticamente liberaciones en el tiempo para unos determinados atributos previamente establecidos, partiendo de la fecha de inicio del despliegue.

Por ejemplo, en AZLinux usamos distintos calendarios (LENTO, NORMAL, RÁPIDO, MUY RÁPIDO) según la criticidad del cambio de software que se va a liberar o de su urgencia. En estos calendarios, asignamos días de demora para los distintos servicios de nuestra organización.

```

CALENDARIO LENTO
  a los 0 días: GRP-EQUIPOS DE TEST.
  a los 5 días: CTX-SERVICIO DE PERSONAL
  a los 10 días: CTX-GESTION TRIBUTARIA
  a los 15 días: SET-ALL SYSTEMS

CALENDARIO MUY RÁPIDO
  a los 0 días: CTX-SERVICIO DE PERSONAL, CTX-GESTION TRIBUTARIA
  a los 2 días: SET-ALL SYSTEMS

```

Es conveniente que en la última demora asignes, si procede, el atributo SET-ALL SYSTEMS.

Cuando asignas un calendario a un despliegue, podrás ver la temporalización resultante en la columna línea temporal de Liberación-Despliegues (pulsas en el desplegable que contiene el nombre del calendario).

Asignar un calendario a un despliegue no es obligatorio.

Esta programación de la liberación se utiliza, fundamentalmente, para conseguir:

- No aplicar una liberación de golpe a muchos equipos, lo que puede provocar un consumo de tráfico de red intenso (imagina 1000 equipos actualizando LibreOffice a la vez).
- Liberar poco a poco los paquetes y así poder hacer comprobaciones más tranquilamente. Cualquier error en el empaquetado o *bug* en los fuentes del paquete, puede ser más manejable si ha afectado a pocos equipos y no a la totalidad.

Un determinado cliente tendrá acceso al despliegue si:

- Tiene un atributo que coincide con alguno de los asignados en el despliegue y ya se ha cumplido la fecha de inicio del despliegue.
- O existe un atributo coincidente con el calendario cuya fecha de inicio del despliegue más la demora se ha cumplido.
- Siempre y cuando un atributo del cliente no coincida con el campo atributos excluidos del despliegue.

Una manera en que puedes ver una estimación de la cantidad de equipos que un calendario va haciendo efectivos los despliegues a lo largo de los días es accediendo a la línea temporal en cada despliegue.

Campos de calendario

- **Nombre:** Denomina al calendario.
- **Descripcion:** Describe el calendario.
- **Demoras:** Es un conjunto de días (demoras) a los que se asignan atributos.
 - **Demora:** Número de días desde la fecha del repositorio a los que los atributos asignados serán efectivos en el repositorio. No se tienen en cuenta ni sábados ni domingos.
 - **Atributos:** Lista de atributos para una demora.
 - **Duración:** Número de días en que se completará el despliegue a los equipos asignados a la demora. O dicho de otra forma, si asignamos el atributo SET-ALL SYSTEMS y una duración de 20 días, obtendríamos un incremento diario aproximado del 5 % del total de equipos.

3.2.7 Repositorios internos vs externos

Llamamos repositorio interno al repositorio que controla el servidor *migasfree*.

Un repositorio externo es un repositorio configurado en los clientes y que no apunta al servidor *migasfree*. Los repositorios que vienen por defecto configurados en las distribuciones son un ejemplo. Otro serían los repositorios tipo ppa.

Si quieres tener un mayor control de tus sistemas, mi recomendación es que te bajes todos los paquetes de los repositorios de tu distribución a una fecha y luego los subas como conjunto de paquetes al servidor y crees un despliegue al efecto. A esto, lo denominamos congelar un repositorio.

De esta manera, tendrás congelados a una fecha los repositorios de tu distribución, y podrás actualizar sólo el software que te interese. Si te decides por este método, obviamente tendrás que empaquetar un código que deshabilite los repositorios externos en los clientes.

Repositorios Internos	Repositorios Externos
Requieren mantenimiento ante las actualizaciones de los paquetes	No requieren mantenimiento ya que es mantenido por el dueño del repositorio
Mayor control de los sistemas frente a los cambios, siendo tu quién decide qué actualizaciones deben producirse	Menor control frente a los cambios
Si el servidor migasfree está en la red local, no produce tráfico internet	Genera tráfico internet

Un pequeño script para obtener los paquetes de los repositorios externos (en este caso para ubuntu-16.04) podría ser:

```
#!/bin/bash

function download(){
    _SERIE_POCKET=$1
    download_repo "$_SERIE_POCKET" "main"
    download_repo "$_SERIE_POCKET" "multiverse"
    download_repo "$_SERIE_POCKET" "restricted"
    download_repo "$_SERIE_POCKET" "universe"
}

function download_repo(){
    _SERVER=http://en.archive.ubuntu.com/ubuntu
    _PKGS=Packages
    _SERIES=$1
    _REPO=$2
    _PATH=`pwd`
    echo "PATH= $_PATH"
    wget $_SERVER/dists/$_SERIES/$_REPO/binary-amd64/$_PKGS.bz2
    bzip2 -d $_PKGS.bz2
    _FILES=`grep "^filename:" $_PKGS| awk '{print $2}'|sort`
    _TARGET=$_SERIES-$_REPO
    echo "$_FILES" > Packages-$_TARGET
    mkdir -p $_TARGET
    cd "$_TARGET"
    for _f in $_FILES
    do
        _file=${_f:6:${#_REPO}}
        _BASE=`basename $_file`
        mkdir -p `dirname $_file`
        echo "Downloading $_SERIES $_f"
        wget -c -t1 $_SERVER/$_f -O $_file
    done
    cd "$_PATH"
    rm $_PKGS
}

download "xenial-security"
download "xenial-updates"
download "xenial-backports"
download "xenial"
```

3.2.8 El proceso de la liberación

Las tareas que debe realizar un liberador son:

- Controlar que no haya paquetes huérfanos, borrando los paquetes antiguos y creando los despliegues adecuados para los nuevos paquetes.
- Decidir qué calendario es conveniente aplicar a cada despliegue.
- Decidir cuándo un despliegue ha terminado de liberarse (se ha cumplido toda la línea temporal) y qué debe hacerse con sus paquetes.

En AZLinux mayoritariamente, y para no tener muchos despliegues activos, estos paquetes los asignamos a otro despliegue (ya existente para este fin) que tiene asignado sólo el atributo `SET-ALL SYSTEMS`. Los despliegues que nos han servido para liberar poco a poco los paquetes son desactivados (no los borramos) para mantener así la historia de lo que se ha ido haciendo.

3.3 La sincronización de los sistemas

Al fin y al cabo, somos lo que hacemos para cambiar lo que somos.

—Eduardo Galeano

En el capítulo anterior, has aprendido a liberar paquetes desde un servidor *migasfree*. Pero, para que se produzca el cambio de software, no basta sólo con liberarlo. Los clientes deben poder acceder a los despliegues, bajarse los paquetes e instalarlos.

En este capítulo vas a centrarte en el cliente *migasfree* para ir conociendo los comandos que tienes a tu disposición.

3.3.1 El proceso de sincronización

Ahora, creo que es buen momento para aprender qué hace `migasfree --update`:

- Envía mensaje de inicio del proceso de sincronización al servidor.
- Envía errores de anteriores ejecuciones. Si los hay, el servidor creará un registro de `Error`.
- Recibe las `Fórmulas` definidas en el servidor.
- Ejecuta dichas `fórmulas` y los resultados son enviados como `Atributos`. El servidor almacena estos atributos y los asocia con el ordenador que se está sincronizando.
- Recibe el código de las `Fallas` y los `Despliegues` a configurar, basándose en los `Atributos` y la fecha actual. Recibe las impresoras que deben estar instaladas. Además, la lista de paquetes a desinstalar e instalar obligatoriamente, también se reciben en este momento del proceso.
- Ejecuta y envía el resultado de las `Fallas`. Si estas se producen, el servidor creará un registro de `Falla` por cada una de ellas.
- Configura los `Despliegues` que el servidor ha dispuesto en función de los `Atributos` del cliente y de la fecha actual.
- Actualiza los metadatos de los repositorios configurados en el sistema. Consiste simplemente en obtener el índice de paquetes actualizado de cada repositorio.
- Desinstala los paquetes obligatorios. Conjunto de paquetes definidos en el campo `Paquetes a desinstalar` de los `Despliegues` efectivos.
- Instala los paquetes obligatorios. Conjunto de paquetes definidos en el campo `Paquetes a instalar` de los `Despliegues` efectivos.

- Actualiza paquetes disponibles. Si en los *Ajustes del cliente migasfree* `Auto_Update_Packages` es `False`, no se producirá esta actualización.
- Envía al servidor el historial de cambios en el software. Es la diferencia de paquetes instalados en el sistema antes y después de desinstalar, instalar y actualizar los paquetes.
- Envía el inventario de software (los paquetes que tiene ahora mismo ese equipo).
- Envía el inventario de hardware periódicamente según `MIGASFREE_HW_PERIOD` de los *Ajustes del servidor migasfree*.
- Envía los errores de ejecución. Si los hay, el servidor creará un registro de `Error`.
- Se añaden y/o eliminan las impresoras.
- Por último, envía un mensaje de proceso finalizado. Cuando el servidor recibe este mensaje añade un registro de `Sincronización` en la base de datos que se emplean para hacer diferentes estadísticas.

Puedes ver una simulación de esta sincronización, accediendo al menú desplegable del identificador del ordenador en la aplicación web. Ver *Ordenadores*.

3.3.2 El comando `migasfree`

La opción del comando `migasfree --update` es, sin lugar a dudas, la más importante. Su sintaxis es:

```
migasfree -u
migasfree --update
```

`migasfree -u` puede usarse conjuntamente con opción `--force-upgrade` para forzar la sincronización de paquetes a pesar que en el ajuste `Auto_Update_Packages` esté asignado a `False`. Consulta el ajuste `Auto_Update_Packages` de los *Ajustes del cliente migasfree*.

```
migasfree -u -a
migasfree --update --force-upgrade
```

Existen otras opciones que pueden hacer más fácil el mantenimiento a los administradores.

En las organizaciones que usan distintos sistemas operativos con sistemas de paquetería diferentes, tanto para buscar, instalar o desinstalar paquetes, los administradores tienen que utilizar los comandos propios del sistema de paquetería. Utilizar las opciones del comando `migasfree` para realizar estas tareas te permite abstraerte del PMS (no tendrás que estar pensando si estás en un sistema basado en Debian o en un Redhat p.e.):

- Para buscar un determinado paquete en los repositorios, utiliza:

```
migasfree -s <texto>
migasfree --search <texto>
```

- Para instalar un determinado paquete, usa:

```
migasfree -ip <paquete>
migasfree --install --package=<paquete>
```

- Para desinstalar un determinado paquete, usa:

```
migasfree -rp <paquete>
migasfree --remove --package=<paquete>
```

Por último, tienes la opción que permite registrar el equipo cliente en el servidor `migasfree` en caso que en el registro Proyecto del servidor el campo `Autoregistrado`, esté desmarcado.

```
migasfree -g
migasfree --register
```

3.3.3 El comando `migasfree-tags`

Puedes ver una explicación de este comando y de su sintaxis en el campo `Etiqueta` de las *Fórmulas* en el capítulo *La configuración del sistema migasfree*.

3.3.4 El comando `migasfree-label`

Consulta el ajuste `MIGASFREE_HELP_DESK` de los *Ajustes del servidor migasfree* donde se describe este comando.

3.4 La Auditoría

Nuestro conocimiento es necesariamente finito, mientras que nuestra ignorancia es necesariamente infinita.

—Karl Raimund Popper

Una vez que liberas los paquetes y que los equipos se van sincronizando, llega el momento de que veas, de manera centralizada, toda la información que se ha ido generando a consecuencia del proceso de sincronización.

Esto es lo que te vas a encontrar en este capítulo.

3.4.1 Alertas

Muestra en lo que debe actuar el administrador para tener un sistema lo más íntegro posible. Ya lo viste en apartado *Comprobaciones* del capítulo *La configuración del sistema migasfree*.

3.4.2 Ordenadores

Accediendo a `Datos-Ordenadores`, verás la lista de ordenadores que se han registrado en el servidor.

Puedes acceder al hardware de un equipo desde la última columna `Producto` de la lista de ordenadores.

También puedes acceder a los **datos** que están **relacionados** con un equipo determinado, mediante el desplegable que hay a la derecha del identificador del equipo. Así, fácilmente podrías ver la cantidad de errores que ha habido en un equipo, sus fallas, migraciones, actualizaciones, ... Además, en este menú desplegable del ordenador, te vas a encontrar con otras opciones interesantes:

- **Sucesos:** Sobre un calendario, desde la fecha de entrada del equipo en el servidor, se muestra la cantidad diaria de sincronizaciones, errores, fallas, migraciones y registros de estado. A más cantidad diaria de sucesos, el color es más oscuro.
- **Simular sincronización:** Te permite simular lo que entregaría el servidor al cliente cuando éste último ejecuta `migasfree -u`. De esta manera, y basándonos en los actuales atributos del ordenador, puedes ver qué despliegues se configurarían en el cliente, qué paquetes se instalarían o eliminarían o qué fallas se ejecutarían.
- **Hardware:** Nos da acceso al hardware del equipo.
- **Etiqueta:** Te muestra una etiqueta que puede ser impresa para pegarla en el ordenador.

Campos de Ordenador

- **Nombre:** Es el nombre del equipo o el especificado en el ajuste `Computer_Name` de los *Ajustes del cliente migasfree*.
- **Proyecto:** El proyecto migasfree del ordenador.
- **Fecha de alta:** Fecha de alta del ordenador en migasfree.
- **Dirección IP:** La dirección IP del equipo en el momento de la sincronización.
- **Dirección IP reenviada:** Útil si el equipo es virtual y queremos saber la IP del ordenador anfitrión.
- **Comentario.**
- **Estado:** Un ordenador puede tener uno de estos seis estados: `Asignado`, `Reservado`, `Desconocido`, `Disponible`, `En reparación` y finalmente `Baja`.

Estos estados los clasificamos de la siguiente manera:

- **ALTA:** Son todos los ordenadores que no tienen el estado `Baja` y los dividimos en productivos e improductivos.
 - **PRODUCTIVOS:** Conjunto de ordenadores en uso. Los diferentes estados dentro de esta categoría son:
 - ◊ `Asignado`. Es el estado por defecto e indica que el ordenador está siendo usado para el propósito propio de tu organización.
 - ◊ `Reservado`. Indica que el ordenador se utiliza para un uso especial, por ejemplo para pruebas.
 - ◊ `Desconocido`. Cualquier otro uso.
 - **IMPRODUCTIVOS:** Ordenadores que no están siendo utilizados y que podrán utilizarse en el futuro. Dentro de los improductivos encontramos los siguientes estados:
 - ◊ `Disponible`. El ordenador está libre y listo para pasar a un estado productivo inmediatamente.

Nota: Cuando un ordenador pasa a estado `Disponible`, se elimina la asignación de todas sus etiquetas y además, si el `CID` del ordenador se ha asignado en algún Repositorio, Conjunto de Atributos, Dispositivo Lógico, Definición de Falla y/o Demora de Calendario también será eliminada dicha asignación automáticamente. Esto evita configuraciones inapropiadas derivadas de cambios de equipos (si no se hiciera de esta manera, el uso antiguo que se le haya dado a un ordenador podría influir en su uso futuro).

◊ `En reparación`. El ordenador está siendo reparado.

- **BAJA:** Tienen el estado `Baja`. Se indica con este estado que el ordenador no se va utilizar nunca más debido a rotura, robo, venta o por cualquier otro motivo.

Nota: Cuando un ordenador pasa a estado `Baja`, se elimina la asignación de todas sus etiquetas y además, si el `CID` del ordenador se ha asignado en algún Repositorio, Conjunto de Atributos, Dispositivo Lógico, Definición de Falla y/o Demora de Calendario también será eliminada dicha asignación automáticamente.

Puedes consultar más información sobre los CID en las *Fórmulas específicas*.

Puedes asignar un estado por defecto a los nuevos ordenadores con el ajuste de servidor: *MIGAS-FREE_DEFAULT_COMPUTER_STATUS*



Figura 2: Estado asignado, reservado, desconocido, disponible, en reparación y baja.

- **Etiquetas:** Lista de Etiquetas asignadas actualmente al ordenador. Para una explicación del funcionamiento de las etiquetas mira los *Categorías de etiquetas*.
- **Dispositivos lógicos:** Impresoras lógicas configuradas en el ordenador.
- **Dispositivo lógico por defecto:** Indica el dispositivo lógico por defecto en el sistema.
- **Usuario:** Último usuario que ha iniciado sesión en el ordenador.
- **Atributos de la sincronización:** Lista de Atributos que se han obtenido como resultado de ejecutar la Fórmulas en el ordenador cliente en el proceso de sincronización.
- **Fecha de inicio de la sincronización:** Fecha en la que comenzó la última sincronización del cliente migasfree.
- **Fecha de fin de la sincronización:** Fecha en la que se finalizó por última vez la sincronización del cliente migasfree.
- **Tiempo de la última actualización:** Tiempo que ha tardado el cliente migasfree en actualizarse.
- **Errores sin comprobar:** Muestra el número de errores pendientes de comprobar del ordenador.
- **Fallas sin comprobar:** Muestra el número de fallas pendientes de comprobar del ordenador.
- **Inventario de software:** Paquetes actuales instalados en el ordenador.
- **Historial de software:** Registro de los paquetes instalados y desinstalados según se van produciendo en el tiempo.
- **Fecha de la última actualización del hardware:** Fecha en que se envió por última vez el hardware.
- **Producto:** Nombre del ordenador incluido en la placa base. Por ejemplo HP ProDesk 600 G1 SFF (C8T89AV)
- **Uuid:** Es el identificador único universal de la placa base del equipo.

Nota: El cliente de migasfree es el encargado de proporcionar este UUID. En caso de no poder obtenerlo porque el fabricante de la placa base no lo ha asignado o por cualquier otro motivo, el cliente proporciona un UUID basado en la MAC de la primera tarjeta de red que encuentre.

- **Máquina:** Indica si es una máquina física o virtual.
- **Procesador:** Nombre del microprocesador. Ejemplo: Intel Core i5-4590 3.30GHz
- **RAM:** Cantidad de memoria RAM.
- **Almacenamiento:** Cantidad de almacenamiento en disco.
- **Discos:** Número de discos.
- **Dirección MAC:** Direcciones MAC del ordenador.

3.4.3 Usuarios

A medida que el cliente de migasfree va ejecutándose en los equipos, el servidor va añadiendo los usuarios que se han autenticado en el entorno gráfico.

Puedes ver la lista de usuarios en [Datos-Usuarios](#)

Campos de Usuario

- **Nombre:** Nombre de la cuenta de usuario para acceder al equipo.
- **Nombre Completo:** Nombre y apellidos del usuario.

3.4.4 Atributos

A medida que se vayan actualizando los equipos, el servidor migasfree irá añadiendo los atributos enviados por los clientes con objeto de que puedas liberar paquetes en función de estos atributos.

Campos de Atributo

- **Fórmula:** Fórmula a la que hace referencia el atributo.
- **Valor:** Identifica el atributo.
- **Descripción:** Describe el atributo.

Una explicación del funcionamiento de los atributos la puedes obtener en el apartado *Fórmulas* de *La configuración del sistema migasfree*.

3.4.5 Etiquetas

Manualmente podrás añadir etiquetas y asignarlas a ordenadores para liberar software en función de estas.

El funcionamiento de las etiquetas ya lo hemos visto en los *Categorías de etiquetas*.

Campos de Etiqueta

- **Categoría de etiqueta:** Hace referencia al tipo de etiqueta.
- **Valor:** Identifica a la etiqueta.
- **Descripción:** Describe la etiqueta.
- **Ordenadores:** Permite asignar ordenadores a la etiqueta.

3.4.6 Errores

Conforme se vayan produciendo errores en los clientes, irán llegando al servidor y serán mostrados en *Alertas*.

Campos de error

- **Comprobado:** Campo que se marcará manualmente cuando se ha comprobado y solucionado el error.
- **Ordenador:** Equipo en el que se ha producido el error.
- **Proyecto:** Es el proyecto que tenía el equipo cuando se produjo el error.
- **Fecha:** Fecha y hora en que se produjo el error.
- **Descripción:** Mensaje que describe el error. Generalmente corresponde a la salida de error del *front-end* del P.M.S.

3.4.7 Fallas

Ya viste el concepto de *Fallas* y cómo se pueden programar en el capítulo *La configuración del sistema migasfree*, así que no me repetiré.

Lo mismo que ocurre con los errores, conforme las fallas se vayan detectando en los clientes, irán apareciendo en el *Alertas*.

Campos de falla

- **Comprobado:** Campo que se marcará manualmente cuando se ha comprobado y solucionado la falla.
- **Ordenador:** Equipo en el que se ha producido.
- **Proyecto:** Es el proyecto que tenía el equipo cuando se produjo la falla.
- **Definición de falla:** Tipo de Falla. Hace referencia al código que ha generado la falla.
- **Fecha:** Fecha y hora en que se produjo la falla.
- **Resultado:** Mensaje que describe la falla. Corresponde a la salida estándar del código de la *Definición de la falla*.

3.4.8 Migraciones

Como hemos visto al principio de este capítulo, los *Ordenadores* se identifican inequívocamente por el *UUID* de la placa base y, además, mantienen un campo *Proyecto* que se corresponde con el ajuste del mismo nombre de los *Ajustes del cliente migasfree*. Ahora bien, en el momento en que el servidor detecta que no corresponde el proyecto que tiene el ordenador en la base de datos del servidor con el que recibe del equipo, el servidor actualiza el registro *Ordenador* y además añade un registro de *Migración*. De esta manera se consigue llevar un histórico de migraciones.

Campos de Migración

- **Ordenador:** Equipo que se ha migrado de proyecto migasfree.
- **Proyecto:** Proyecto migasfree.
- **Fecha:** Fecha y hora en que se ha detectado el cambio de proyecto.

3.4.9 Notificaciones

Ante hechos relevantes en el sistema, el servidor genera notificaciones para alertar a los administradores.

Campos de Notificación

- **Fecha:** Fecha y hora en que se ha generado la notificación.
- **Notificación:** Describe el hecho.
- **Comprobado:** Campo que se marcará manualmente cuando se ha recibido la notificación.

3.4.10 Consultas

Aquí podrás ejecutar las Consultas disponibles.

Puedes añadir nuevas consultas o modificar las predeterminadas accediendo a Configuración-Consultas. Una pequeña explicación de cómo se programan la puedes encontrar en el apartado *Consultas* de *La configuración del sistema migasfree*.

3.4.11 Estadísticas

Es una lista con estadísticas predefinidas.

- **cuadro de mando:** Resumen de las estadísticas más importantes sobre el parque de ordenadores gestionado por el servidor migasfree.
- **ordenadores actualizados/hora:** Gráfica de barras que indica la cantidad (única) de equipos que han completado la actualización de migasfree por hora.
- **ordenadores actualizados/día:** Gráfica de barras que indica la cantidad (única) de equipos que han completado la actualización de migasfree por día.
- **ordenadores actualizados/mes:** Gráfica de barras que indica la cantidad (única) de equipos que han completado la actualización de migasfree por mes.

3.4.12 El proceso de las comprobaciones

Al igual que como liberador debes realizar un conjunto de tareas para mantener el sistema en codiciones, continuamente te llegarán errores, fallas, etc. que debes comprobar y atender. Esta es la misión para un usuario checker.

¿Qué tareas tienes que hacer como comprobador del sistema? Sencillo. Mantén las Alertas a 0. El sistema te irá avisando qué debes atender.

- Comprueba periódicamente la existencia de Errores. Soluóionalos y márcalos como comprobados.
- Comprueba periódicamente la existencia de Fallas. Soluóionalas y márcalas como comprobadas.
- Comprueba periódicamente la existencia de Notificaciones. Una vez leídas, márcalas como comprobadas.

3.4.13 Otros procesos

Reemplazo de ordenadores

Este proceso permite intercambiar el estado, etiquetas, dispositivos y atributos CID asignados en el sistema entre dos ordenadores.

Imagina que un usuario te reporta un fallo de hardware y decides darle un equipo que tienes en estado disponible para que continúe su trabajo. En este caso ve al menú Datos - Reemplazo de ordenadores e introduce los dos ordenadores. Una vez pulses en el botón Reemplazar el ordenador que estaba disponible tendrá ahora el

estado, etiquetas e impresoras que tenía el ordenador estropeado. Finalmente puedes editar el ordenador que ha fallado y cambiarle el estado a `en reparación` o a `baja`.

3.5 Dispositivos

La lógica es clara y sencilla: la propaganda es a la democracia lo que la cachiporra al estado totalitario.

—Noam Chomsky

Migasfree puede ser utilizado también para instalar de manera centralizada las impresoras, ahorrándote un tiempo más que considerable respecto a una instalación manual en cada ordenador (aunque inicialmente se requiera introducir en el sistema los distintos modelos de impresoras que vayas a utilizar).

Además, por ser un sistema centralizado, te va a permitir conocer el parque de impresoras de tu organización.

En migasfree las impresoras son asignadas a atributos, por lo que puedes especificar que una determinada impresora sea instalada en todos los ordenadores que tengan, por ejemplo, la etiqueta `AUL-AULA_DE_FORMACION` aunque, obviamente, lo más común será asignarlas a un ordenador concreto utilizando su CID (Computer ID).

Nota: La instalación de las impresoras se producirá cuando un cliente haga `migasfree -u` ya que el servidor indica en ese momento al cliente qué impresoras debe tener instaladas. El cliente es el que con esa información instala, desinstala o actualiza las impresoras.

3.5.1 Pasos para la configuración de las impresoras:

- Añadir los **fabricantes**.
- Definir las diferentes **prestaciones** que vas a utilizar.
- Añadir los **modelos** y crear un paquete por cada modelo de impresora.
- Añadir cada **dispositivo** físico, especificando cómo está conectado y qué dispositivos lógicos deben instalarse a cada `atributo`.

3.5.2 Fabricantes

Campos del Fabricante

- **Nombre:** Nombre del fabricante de la impresora.

3.5.3 Prestaciones

Representa una determina configuración por defecto del controlador del dispositivo.

Esto sirve para facilitar a los usuarios la impresión, ya que por cada impresora física puedes querer instalarle diferentes configuraciones por defecto. Por ejemplo, puedes querer configurar estas 3 impresoras lógicas:

- Impresión en negro, doble cara y calidad borrador
- Impresión en color y en 1 cara
- Impresión usando la bandeja multipropósito

Así que en este caso te interesaría crear las siguiente prestaciones:

- BORRADOR
- COLOR
- MULTIPROPOSITO.

Campos de la Prestación

- **Nombre:** Nombre la prestación.

3.5.4 Modelos

Por cada modelo de impresora se especifican las distintas conexiones y los distintos controladores que estarán admitidos.

Campos del modelo de dispositivo

- **Nombre:** Nombre del modelo de la impresora.
- **Fabricante:** Asigna el Fabricante del modelo de impresora.
- **Tipo:** Actualmente sólo se utiliza “PRINTER”.
- **Conexiones:** Especifica las distintas maneras en las que puedes conectar físicamente la impresora.
- **Controladores:** Por cada **version** y **prestación** tendrás que definir que archivo ppd debe ser utilizado.

Un archivo PPD (PostScript Printer Description) describe las características disponibles para la impresora. Dicho archivo puede ser obtenido directamente del fabricante o en su defecto de openprinting.org

- **Proyecto:** Proyecto migasfree.
- **Prestación:** Prestación.
- **Nombre:** Ruta completa donde esta ubicado el archivo .ppd en los ordenadores.
- **Paquetes a instalar:** En este campo deberás poner el nombre del paquete que contiene el archivo ppd que vas a utilizar. Si es necesario algún otro paquete, puedes añadirlo separándolo con un espacio.

3.5.5 Dispositivos

Representa a cada una de las distintas impresoras físicas.

Campos del dispositivo

- **Nombre:** Identificador único de la impresora física (en AZLinux usamos el nº de inventario que lleva impreso la etiqueta que pegamos en cada impresora).
- **Modelo:** Modelo de la impresora
- **Conexión:** Como se conecta físicamente la impresora.
- **Dispositivos lógicos:**
 - **Prestación:** Prestación.
 - **Nombre:** Nombre alternativo a la prestación. Es opcional.

- **Atributos:** Atributos a los que será instalado el dispositivo lógico.

Nota: La forma más eficaz de añadir los dispositivos es asignar el nombre, modelo y conexión y entonces pulsar en `grabar` y `continuar` editando. De esta manera se añadirán automáticamente los dispositivos lógicos asociados a ese modelo de impresora y podrán ser asignados entonces los atributos.

3.5.6 Reemplazo de dispositivos

Imagina p.e. que tienes una impresora de red instalada en 30 ordenadores y que va a ser sustituida por otro modelo de impresora. En este caso, lo único que tendrás que hacer es un `Reemplazo de dispositivos` indicando el dispositivo antiguo y el nuevo. Solamente eso.

Si has leído y entendido bien este último párrafo, deberías ya comprender cómo `migasfree` puede ahorrarte mucho tiempo en todo lo relacionado a la instalación de impresoras.

3.5.7 Ejemplo EPSON AL-M300

Para afianzar los conceptos vamos a crear el modelo `EPSON AL-M300` y a instalar dos impresoras físicas con estas tres prestaciones:

- **BN:** Impresión en negro.
- **MP:** Bandeja multipropósito.
- **DUPLEX:** Doble cara

Para ello descárgate los ejemplos de este libro donde encontrarás una carpeta llamada `acme-epson-al-m300`

```
$ wget https://github.com/migasfree/fun-with-migasfree-examples/archive/master.zip
$ unzip master.zip
$ cd fun-with-migasfree-examples-master/acme-epson-al-m300
```

Observa que en el paquete existen tres archivos `.ppd` que se corresponden con las prestaciones comentadas.

```
$ ls /usr/share/ppd/acme
Epson_AL_M300-duplex-ps-es.ppd
Epson_AL_M300-MP-ps-es.ppd
Epson_AL_M300-ps-es.ppd
```

Una manera simple de obtener estos archivos `ppd`, es la siguiente:

1. Obtén el `ppd` del propio fabricante de la impresora o en su defecto de openprinting.org
2. Instala una impresora en tu sistema con el archivo `ppd` obtenido.
3. **Modifica las propiedades** de la impresora desde el interface gráfico de usuario de acuerdo a la prestación que va a proporcionarte (ejecuta p.e. `system-config-printer` en un terminal).
4. En el directorio `/etc/cups/ppd/` tendrás el `ppd` con las opciones que has elegido para tu prestación, cópialo con un nombre relacionado con esa prestación.

Crea ahora el paquete (debes tener el paquete `devscripts` y `debhelper` previamente instalados).

```
$ /usr/bin/debuild --no-tgz-check -us -uc
```

Súbelo al servidor

```
# migasfree-upload -f ../acme-epson-al-m300_1.0-1_all.deb
```

y libéralo creando un nuevo repositorio en el servidor llamado p.e. impresoras. Asígnale el paquete `acme-epson-al-m300_1.0-1_all.deb` y en atributos asigna `ALL-SYSTEMS`.

Ve a `Dispositivos - Fabricantes` y añade `EPSON`.

En `Dispositivos - Prestaciones` añade `BN MP` y `DUPLEX`

Añade un `Dispositivo - Modelos` de la siguiente manera:

- Nombre: `AL-M300`
- Fabricante: `EPSON`
- Tipo: `PRINTER`
- Conexiones: `USB` y `TCP`. (Aquí especificamos las distintas conexiones que tiene la impresora)

Pulsa en `Grabar` y continuar editando.

Ahora añade los controladores:

- Controlador 1:
 - Proyecto: `debian-8.6`. (u otra)
 - Prestación: `BN`
 - Nombre: `/usr/share/ppd/acme/Epson_AL_M300-ps-es.ppd`
 - Paquetes a instalar: `acme-epson-al-m300`
- Controlador 2:
 - Proyecto: `debian-8.6`. (u otra)
 - Prestación: `MP`
 - Nombre: `/usr/share/ppd/acme/Epson_AL_M300-MP-ps-es.ppd`
 - Paquetes a instalar: `acme-epson-al-m300`
- Controlador 3:
 - Proyecto: `debian-8.6`. (u otra)
 - Prestación: `DUPLEX`
 - Nombre: `/usr/share/ppd/acme/Epson_AL_M300-duplex-ps-es.ppd`
 - Paquetes a instalar: `acme-epson-al-m300`

Hasta aquí hemos definido los modelos de impresoras. Ahora ya estamos en disposición de añadir las impresoras a los equipos.

Vamos a añadir 2 impresoras físicas de ejemplo al ordenador `CID-1`.

Impresora Física 1 conectada por TCP

Ve a `Dispositivos - Dispositivos` y añade la primera impresora:

- Nombre: `Impresora1`
- Modelo: `AL-M300`
- Conexión: Elije `TCP`

- IP: 10.0.2.250
- Pulsa ahora en `Grabar` y continuar editando y aparecerán los tres dispositivos lógicos correspondientes. Deja el `Nombre` en blanco. Añade a cada uno de los dispositivos lógicos el atributo `CID-1` (o el atributo que quieras).

Impresora Física 2 conectada por USB

Ve a `Dispositivos - Dispositivos` y añade la segunda impresora:

- Nombre: `Impresora2`
- Modelo: `AL-M300`
- Conexión: `Elije ahora USB`.
- Pulsa ahora en `Grabar` y continuar editando y aparecerán los tres dispositivos lógicos correspondientes. Deja el `Nombre` en blanco. Añade a cada uno de los dispositivos lógicos el atributo `CID-1` (o el atributo que quieras).

Y con esto, al hacer `migasfree -u` en cada ordenador se instalarán las impresoras lógicas según hemos especificado.

Para finalizar, conviene mencionar que cualquier cambio que se realice en el dispositivo (la IP, el tipo de conexión, o el nombre p.e.) será modificado automáticamente en los ordenadores cuando estos se actualicen.

3.6 Ejemplos prácticos

La imaginación tiene sobre nosotros mucho más imperio que la realidad.

—Jean de la Fontaine

A continuación, te voy a mostrar una selección representativa de peticiones e incidencias de la Gestión de Configuración Software, junto con una solución, la que creo más acertada a día de hoy, porque es obvio que no existe una única solución a los problemas.

El objetivo de estos ejemplos (basados en casos reales del proyecto `AZLinux`) es afianzar los conceptos que hemos ido viendo durante los capítulos anteriores. De esta manera, podrás hacerte una mejor idea de toda la potencia de trabajar con la paquetería estándar de tu distribución Linux junto con `migasfree`, en el proceso de la GCS.

El código fuente de los paquetes utilizados en este capítulo está disponible en <http://github.com/migasfree/fun-with-migasfree-examples>

Descárgatelo si aún no lo has hecho:

```
$ wget https://github.com/migasfree/fun-with-migasfree-examples/archive/master.zip
$ unzip master.zip
$ cd fun-with-migasfree-examples-master
```

3.6.1 Deshabilitar montaje de unidades USB y CDRM

a) Petición de cambio

Todos los ordenadores de la `NET-192.168.100.0/24`, excepto el ordenador `CID-3245`, no deberían montar unidades USB por motivos de seguridad. Tampoco deben montar unidades USB aquellos usuarios que pertenezcan al grupo de LDAP `GRP-NO-MEDIA`, ni tampoco el ordenador `CID-1291`.

b) Aceptación de la petición

Realizaremos lo que denominamos una directiva de seguridad basada en un conjunto de atributos, ya que es muy probable que, con el tiempo, esta medida de seguridad puede ir cambiando sus reglas asignándose o excluyéndose otros atributos.

c) Cambio

Después de buscar en Internet la forma de bloquear el montaje de dispositivos USB, y de varias pruebas, llegamos a la conclusión que realizar un cambio de permisos en el directorio `/media` es suficiente para alcanzar nuestro objetivo.

ECS: acme-media-disable (nuevo)

Decidimos crear el paquete `acme-media-disable` que bloqueará el montaje en `/media` cuando esté instalado este paquete.

Observa como en la **post-instalación** del paquete se restringe los permisos a `/media`, y como en la **post-desinstalación** dejamos los permisos por defecto.

Crema el paquete `acme-media-disable` (debes tener previamente instalados el paquete `devscripts` y `debhelper`)

```
$ cd acme-media-disable
$ /usr/bin/debuild --no-tgz-check -us -uc
$ cd ..
```

y súbelo al servidor.

```
# migasfree-upload -f acme-media-disable_1.0-1_all.deb
```

d) Liberación

Creamos el conjunto de atributos `SET-NO-MEDIA`:

ATRIBUTOS:

`NET-192.168.100.0/24`

`GRP-NO-MEDIA`

`CID-1291`

EXCLUYE:

`CID-3245`

Creamos un despliegue `MEDIA-DISABLE`:

Asignamos el paquete: `acme-media-disable_1.0-1_all.deb`

Ponemos en paquetes a instalar: `acme-media-disable`

asignamos en atributos incluidos: `SET-NO-MEDIA`

Ahora es necesario crear otro despliegue, `MEDIA-ENABLE`, para forzar la desinstalación del paquete cuando el ordenador ya no pertenezca al conjunto de atributos `SET-NO-MEDIA`:

Ponemos en paquetes a desinstalar: `acme-media-disable`

asignamos en atributos incluidos: `ALL-SYSTEMS`

asignamos en atributos excluidos: `SET-NO-MEDIA`

Para añadir o excluir atributos en el futuro, simplemente bastará con modificar el conjunto de atributos `SET-NO-MEDIA`.

Nota: Este es un ejemplo de cómo implementar una «directiva de seguridad». En versiones futuras de migasfree se incluirá un modelo de «directivas» para no tener que añadir ese segundo despliegue que fuerza la desinstalación de los paquetes.

3.6.2 Cierre de sesión gráfica en ordenadores HP ProDesk 600 G2 SFF

a) Petición de cambio

Aleatoriamente y de repente, se cierra la sesión gráfica de usuario en los equipos HP ProDesk 600 G2 SFF

Frecuencia aproximada del error: de 0 a 3 cierres de sesión por día.

Se han recibido incidencias de este tipo sólo en algunos equipos de este modelo de ordenador.

b) Aceptación

Se comprueba, observando el fichero `/var/log/syslog`, que el error en estos equipos es provocado por la tarjeta gráfica `intel-hd-graphics-530`, que se identifica como `PCI-8086:1912` incluida en los ordenadores HP ProDesk 600 G2 SFF.

c) Cambio

Después de buscar en Internet información relativa a este error, y de probar distintas soluciones, llegamos a comprobar que modificando el método de aceleración de la tarjeta gráfica, siguiendo un [workaround](#) propuesto en Internet, el error ya no se reproduce.

ECS: `acme-intel-hd-graphics-530` (nuevo)

Creamos un paquete `acme-intel-hd-graphics-530` que, simplemente, modifica el método de aceleración añadiendo en el sistema el fichero `/usr/share/X11/xorg.conf.d/20-intel-hd-graphics-530.conf` con el siguiente contenido:

```
Section "Device"
    Identifier "Card0"
    Driver "Intel"
    Option "AccelMethod" "uxa"
EndSection
```

y lo subimos al servidor.

d) Liberación

Dado que tenemos en producción unos 120 equipos con la tarjeta gráfica `PCI-8086:1912` desplegaremos este paquete poco a poco (sólo a los ordenadores que tengan dicho atributo) ya que no sabemos, a priori, qué otros efectos no deseados puede provocar el cambio de aceleración gráfica.

Creamos un calendario únicamente para este despliegue:

Calendario `intel-hd-graphics-530`

Demora: 0

Atributos: `PCI-8086:1912`

Duración: 20 días

Creamos el despliegue `intel-hd-graphics-530`

Ponemos en paquetes a instalar: `acme-intel-hd-graphics-530`

Asignamos en calendario: `intel-hd-graphics-530`

Nota: De esta manera se actualizarán aproximadamente $120/20 = 6$ ordenadores por día, lo que puede ser manejable por el servicio telefónico de Asistencia a Usuarios en caso de producirse efectos no deseados.

3.6.3 Etiqueta MEDIA-DISABLE en migasfree

a) Petición de cambio

Crear una etiqueta en migasfree para desplegar software en los equipos que no deben montar unidades USB y CDROM.

b) Aceptación

Se rechaza la petición, debido a que ya disponemos del conjunto de atributos `SET-MEDIA-DISABLE`.

Nota: Los conjuntos de atributos son mucho más **versátiles** que una simple etiqueta asignada a un ordenador. La etiqueta está pensada para que un usuario puede asignarla desde su ordenador mediante el comando `migasfree-tags`. En cambio, los conjuntos de atributos son definidos en el servidor por un administrador mediante la asignación y exclusión de atributos y/o otros conjuntos de atributos.

3.6.4 Instalación de software en AULA-3

a) Petición de cambio

Se va a proceder a impartir cursos sobre `gimp` en el AULA-3. Es necesario que este software esté instalado esta misma tarde, a las 16:00 h, en todos los ordenadores de dicha aula.

b) Aceptación

Existe un despliegue `per-aula-3` para añadir software a los equipos de dicha aula. Como atributos incluidos, este despliegue tiene el conjunto de atributos `SET-AULA-3`

Conjunto de atributos `SET-AULA-3`:

`CID-3578`

`CID-3579`

`CID-3580`

`CID-3581`

`CID-3582`

`CID-3583`

`CID-3584`

`CID-3585`

Despliegue `per-aula-3`:

paquetes a instalar:

`inkscape`

`scribus`

`gvsig`

`virtualbox`

atributos incluidos: `SET-AULA-3`

c) Cambio

En este caso no hay implicado ningún ECS que tenga que ser modificado.

d) Liberación

Editamos el despliegue `per-aula-3`

añadimos a los paquetes a instalar : `gimp`

Nota: Cuando los equipos del AULA-3 inicien la próxima sesión gráfica, automáticamente se les instalará el software solicitado. Observa que no es necesario desplazarnos al aula, acceder por control remoto, ni tan siquiera encender los equipos. El software se instalará cuando, plácidamente, esté echando mi siesta después de comer.

Puesta en producción

La sublimación de los bárbaros.

4.1 Migasfree en producción

El valor del producto se halla en la producción.

—Albert Einstein

Deberás disponer de una máquina con un S.O. linux de 64 bits sobre la que vas a instalar docker y poder ejecutar los 2 contenedores necesarios, uno es para la BD y el otro para la aplicación.

Nota: Desde la versión 4.14 del servidor migasfree sólo proveemos contenedores `docker` para ejecutarlo. Esto nos ha liberado de la tediosa tarea de tener que construir los paquetes necesarios para su ejecución, a la vez que nos aprovechamos de las beneficios de usar contenedores.

Nota: Importante: Asegúrate de cambiar las contraseñas a los usuarios que vienen por defecto.

4.1.1 Instalación

Sigue los pasos indicados en `migasfree-docker`

4.1.2 Mantenimiento de contenedores

Si nunca has usado docker te describo las instrucciones más básicas para que puedas empezar a manejarte con docker.

- `docker images #` muestra las imágenes.
- `docker ps -a #` muestra los contenedores.

- `docker rmi -f <imagen> # borra una imagen.`
- `docker rm -f <contenedor> # borra un contenedor.`
- `docker restart <contenedor> # reinicia un contenedor.`
- `docker exec -ti <contenedor> bash # ejecuta un bash interactivo dentro de un contenedor.`

4.1.3 Actualización del servidor migasfree

Puedes actualizar el servidor modificando la variable `MIGASFREE_VERSION` del fichero `variables` y ejecutando de nuevo:

```
# . variables
# docker-compose up -d
```

Nota: Puedes ver las distintas versiones disponibles en el [repositorio de docker](#)

4.1.4 Configuración del servidor

Una vez en funcionamiento el servidor puedes configurar el servidor mediante el fichero `/var/lib/migasfree/FQDN/conf/settings.py`

Hay diversas variables que se pueden configurar aquí para modificar el comportamiento de migasfree. Para una personalización más avanzada, mira los *Ajustes del servidor migasfree*.

No olvides reiniciar el contenedor si modificas alguno de estos ajustes.

Cambiando las contraseñas

- Accede a `Configuracion-Usuarios` y verás los perfiles de usuarios.
- Edita el usuario `admin`. Cámbiale la contraseña y guárdalo.
- Edita el resto de perfiles de usuario y deshabilítalos o cámbiales la contraseña.

La explicación de los usuarios y sus grupos lo has visto ya en *La configuración del sistema migasfree*

4.1.5 Servicio de caché de paquetes

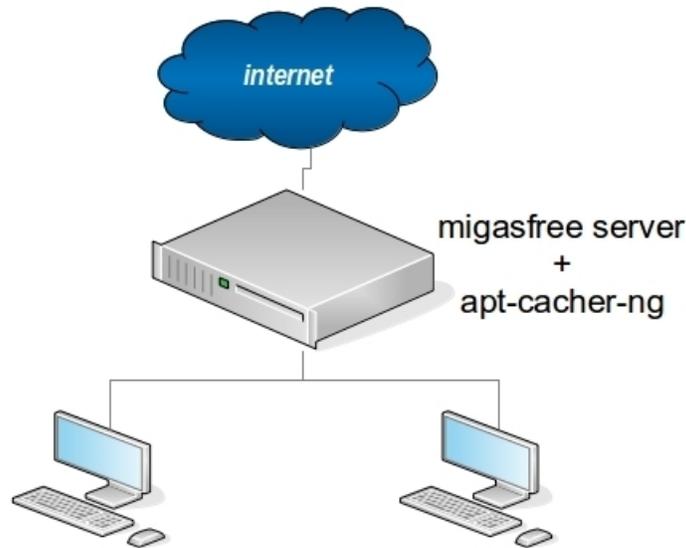
Montar un caché de paquetes para disminuir el tráfico de Internet es habitual. Su funcionamiento es muy sencillo. Cuando un equipo necesita descargar un paquete de Internet, lo solicita al caché. Si el servicio de caché no lo tiene ya almacenado, lo descargará de Internet, lo almacenará y se lo ofrecerá al equipo. Si otro equipo necesita ese mismo paquete, como ya está en la caché ya no se producirá tráfico Internet, sino que el servicio de caché lo ofrecerá directamente al equipo.

Puedes instalar el servicio de caché de paquetes en el equipo donde has instalado el servidor migasfree, o en otro servidor.

Por ejemplo puedes instalar `apt-cacher-ng`.

```
# apt-get install apt-cacher-ng
```

Configura el usuario para la administración del servicio.



```
# nano /etc/apt-cacher-ng/security.conf
```

Descomenta la línea que empieza por AdminAuth y modifica el usuario y la contraseña:

```
AdminAuth: <usuario>:<contraseña>
```

Reinicia el servicio.

```
#service apt-cacher-ng restart
```

Por defecto el puerto del servicio `apt-cacher-ng` es el **3142**. Accede a la página `http:<miservidor>:3142` para la administración del servicio de caché.

Hasta aquí hemos instalado y configurado el caché en el servidor.

Para la configuración de los clientes, debes crear el fichero `/etc/apt/apt.conf.d/02proxy` con el siguiente contenido:

```
Acquire::http { Proxy "http://<miservidor>:3142"; };
```

Para hacerlo correctamente, modifica el paquete `acme-migasfree-client` añadiendo este fichero al paquete.

Otra manera de configurar los clientes es haciendo uso del ajuste `Package_Proxy_Cache` de los *Ajustes del cliente migasfree*. La diferencia entre éste método y el anterior es que el primero hará uso del servicio del caché de paquetes tanto cuando ejecutes el comando `migasfree` en los clientes, como cuando ejecutes el gestor de paquetes (`apt-get`). En cambio, en el segundo método sólo usará el servicio de caché al ejecutar el comando `migasfree`.

Puede consultar el [manual de apt-cacher-ng](#) para una configuración más avanzada del servicio de caché.

4.1.6 Etiquetando los clientes

Para facilitar la atención a los usuarios cuando tengan un problema, es conveniente imprimir y pegar físicamente la etiqueta que identifica inequívocamente a cada equipo, ejecutando desde el cliente el comando:

```
migasfree-label
```

Consulta el ajuste `MIGASFREE_HELP_DESK` de los *Ajustes del servidor migasfree*

Nota: También puedes imprimir la Etiqueta desplegando el menú del ordenador en el servidor.

4.2 Creando tu propia Distro

No tratéis de guiar al que pretende elegir por sí su propio camino.

—William Shakespeare

Una *Distro* no es más que un conjunto de software seleccionado y preparado para instalarse fácilmente.

Existen [herramientas](#) que te permiten personalizar una distribución Linux fácilmente sin grandes complicaciones, y también puedes crear tu [Linux desde cero](#), eso sí, armándote de paciencia.

Pero como te decía, para crear tu Distribución personalizada debes:

- Seleccionar el software que incluirás en ella.
- Preparar un sistema sencillo para instalar todo ese software.

En este capítulo te describo diferentes opciones que he utilizado para realizar estas dos tareas.

4.2.1 La selección de paquetes

La idea principal al trabajar con migasfree es que todo debe ser empaquetado, incluida la personalización del software. Así que debes elegir qué software incluirá tu distribución y crear los paquetes que la personalicen.

El método que te expongo a continuación es muy versátil, y es el de utilizar migasfree para especificar esta selección de paquetes.

Se trata de usar los campos del Despliegue:

- Paquetes pre-incluidos por defecto. Lista de paquetes que configuran repositorios externos.
- Paquetes incluidos por defecto: Lista de paquetes a instalar.
- Paquetes excluidos por defecto: Lista de paquetes a desinstalar.

y asignar al Despliegue el Atributo incluido `SET-ALL SYSTEMS`

No tienes por qué indicar todos los paquetes. Como vamos a partir de una distribución generalista como Debian, Ubuntu, RedHat, etc. indicamos sólo los paquetes que queremos añadir o eliminar a la *Distro*.

También puedes hacer una selección de los paquetes que compondrían unos «sabores», y en vez de usar el Atributo `SET-ALL SYSTEMS`, crear una Etiqueta por sabor y asignarla en diferentes Despliegues.

La creación de etiquetas la viste en [La configuración del sistema migasfree](#).

4.2.2 La instalación de tu Distribución

Ahora que has elegido y creado uno o varios Despliegues en migasfree con los paquetes que debe llevar tu distribución, es el momento de ver varios métodos para instalar tu distribución personalizada y controlada desde un servidor migasfree.

El método de «andar por casa»

Recomendado si no quieres complicarte la vida y tienes pocos equipos en los que instalar tu *Distro*:

1. Instala la distribución generalista en el equipo.
2. Instala y configura el cliente `migasfree`. Si es preciso registra el ordenador mediante `migasfree --register`.
3. Ejecuta `migasfree-tags --set`.

Generando un Live/CD

Es el método recomendado si la instalación la puede realizar cualquier persona. Se trata de hacer básicamente lo mismo que en el método anterior pero substituyendo el primer paso por el empleo de un Live/CD en un entorno «chroot».

1. Prepara un entorno chroot con el Live/CD de partida.
2. Instala y configura el cliente `migasfree` en el entorno chroot. Si es preciso registra el ordenador mediante `migasfree --register`.
3. Ejecuta `migasfree-tags --set` dentro del entorno.
4. Finalmente genera una imagen ISO del entorno.

Puedes ver un ejemplo de cómo se hace en `vitalinux` con `vx-create-iso`.

Clonación de imagen

Es el método que usamos en AZLinux y está recomendado si tienes muchos equipos y eres tú quien hace las instalaciones.

1. Instala la distribución generalista en un equipo que hará de «máster».
2. Instala y configura el cliente `migasfree`. Si es preciso registra el ordenador mediante `migasfree --register`.
3. Ejecuta `migasfree-tags --set`.
4. Prepara el equipo para clonar y crea una imagen para su clonación.
5. Clona la imagen en los equipos.

4.2.3 Reinstalando la selección de paquetes

Una vez instalada tu *Distro*, en cualquier momento podrás cambiar de sabor ejecutando otra vez `migasfree-tags --set`. Observa que has podido decidir cambiar la selección de paquetes en `migasfree` entre tanto, esto te permite ir probando tu *Distro* fácilmente mientras aún la estás definiendo.

4.3 FAQ

Uno reconoce a las personas inteligentes por sus respuestas. A los sabios se los reconoce por sus preguntas.

—Naguib Mahfuz

4.3.1 Sobre el servidor migasfree

Quando accedo al servidor web me aparece: **Server error (500)**

Causa

Este error puede estar motivado por múltiples causas. La más probable es que la contraseña del usuario migasfree en Posgresql no sea la misma que la que está configurada en el servidor.

Solución

Comprueba la contraseña que tienes en `/var/lib/migasfree/FQDN/conf/settings.py` es la misma que la del usuario migasfree en Postgresql. Si no existe este fichero, la contraseña por defecto es migasfree.

Si necesitas cambiarla, haz esto:

```
# su postgres
# psql
# ALTER USER migasfree WITH PASSWORD 'mipassword';
```

¿Cómo migro el servidor a una versión \geq 4.14 con docker?

En primer lugar sigue los pasos indicados en *Migasfree en producción*.

Observa ahora que en el fichero `docker-compose.yml` se establecen para el servidor los volumes siguientes:

```
volumes:
- "/var/lib/migasfree/${FQDN}/conf:/etc/migasfree-server"
- "/var/lib/migasfree/${FQDN}/public:/var/migasfree/repo"
- "/var/lib/migasfree/${FQDN}/keys:/usr/share/migasfree-server"
```

Por tanto, deberás mover o copiar los ficheros de la versión antigua a `/var/lib/migasfree/${FQDN}/` además de cambiar a estos ficheros el propietario (el servidor \geq 4.14 utiliza un usuario con uid y gid 890).

```
# cp /etc/migasfree-server/* /var/lib/migasfree/${FQDN}/conf
# cp /var/migasfree/repo/* /var/lib/migasfree/${FQDN}/public
# cp /usr/share/migasfree-server/* /var/lib/migasfree/${FQDN}/keys
# chown -R 890:890 /var/lib/migasfree/${FQDN}/public
# chown -R 890:890 /var/lib/migasfree/${FQDN}/keys
```

Por último, es muy recomendable que regeneres los metadatos de los despliegues. Esto evitará que se produzcan errores en el cliente de firmas GPG inválidas, al no estar firmado el fichero `InRelease` en los proyectos con sistema de paquetería `apt`, al haberse generado los metadatos de los repositorios con versiones antiguas del servidor:

```
Get:1 http://migasfree/repo/Ubuntu-18.04/REPOSITORIES test InRelease [4213 B]
Err:1 http://migasfree/repo/Ubuntu-18.04/REPOSITORIES test InRelease
  The following signatures were invalid: 37CDCDA342A718EADA493BC5827CFFCB9A18B812
Hit:2 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
W: GPG error: http://migasfree/repo/Ubuntu-18.04/REPOSITORIES test InRelease: The
↳ following signatures were invalid: 37CDCDA342A718EADA493BC5827CFFCB9A18B812
E: The repository 'http://migasfree/repo/Ubuntu-18.04/REPOSITORIES test InRelease' is
↳ not signed.
```

(continues on next page)

(proviene de la página anterior)

```
N: Updating from such a repository can't be done securely, and is therefore disabled
↳by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
```

Para ello accede a Liberación - Despliegues, selecciona los despliegues que necesitas regenerar, en el desplegable acción elige regenerar metadatos y pulsa finalmente sobre el botón ir

¿Cómo hago una fórmula para obtener el contexto LDAP de un usuario?

Necesitas que los clientes tengan instalado el paquete python-ldap. En el servidor tendrás que crear una nueva fórmula:

Prefijo: CTX

Nombre: CONTEXTO LDAP

Lenguaje: python

Clase: Añadir por la derecha

Código:

```
import sys
import ldap
import migasfree_client.utils

LDAP_SERVER = 'ldap.miservidor.es'
LDAP_BASE = ''
LDAP_SCOPE = ldap.SCOPE_SUBTREE

def get_ldap_property(filter_str, property_str, base = LDAP_BASE, scope = LDAP_SCOPE):
    global global_ldap_object

    try:
        _result = global_ldap_object.search_s(base, scope, filter_str, [property_str])
    except ldap.LDAPError, e:
        print e
        sys.exit(errno.ENOMSG) # no result

    if _result == None or not _result:
        print 'No result in LDAP search'
        sys.exit(errno.ENOMSG) # no result

    if property_str == 'dn': # special case: dn is gotten in other field
        return _result[0][0]

    try:
        _ret = _result[0][1][property_str]
        if len(_ret) == 1: # only one result?
            return _ret[0]
    except KeyError:
        return '' # empty value

    return _ret

def get_dn(user):
    # cn=oXXXXX,ou=XXXX,o=XXXXXX
    return get_ldap_property('(cn=%s)' % user, 'dn')

def get_context(user):
    result = get_dn(user).split(',')

    ret = ''
```

(continues on next page)

(proviene de la página anterior)

```
for item in result[:]:
    tmp = item.split('=')
    if tmp[0] == 'ou' or tmp[0] == 'o':
        ret = '%s%s.' % (ret, tmp[1])

return ret[:-1] # remove trailing '.'

def run():
    global global_ldap_object
    global_ldap_object = ldap.initialize('ldap://%s:389' % LDAP_SERVER)

    user=migasfree_client.utils.get_current_user().split("~")[0]
    print get_context(user)

if __name__ == '__main__':
    run()
```

¿Cómo hago una fórmula para obtener los grupos LDAP de un usuario?

Necesitas que los clientes tengan instalado el paquete python-ldap. En el servidor tendrás que crear una nueva fórmula:

Prefijo: GRP

Nombre: GRUPOS LDAP

Lenguaje: python

Clase: Lista

Código:

```
import sys
import ldap
import migasfree_client.utils
LDAP_SERVER = 'ldap.miservidor.es'
LDAP_BASE = ''
LDAP_SCOPE = ldap.SCOPE_SUBTREE

def get_ldap_property(filter_str, property_str, base = LDAP_BASE, scope = LDAP_SCOPE):
    global global_ldap_object

    try:
        _result = global_ldap_object.search_s(base, scope, filter_str, [property_str])
    except ldap.LDAPError, e:
        print e
        sys.exit(errno.ENOMSG) # no result

    if _result == None or not _result:
        print 'No result in LDAP search'
        sys.exit(errno.ENOMSG) # no result

    if property_str == 'dn': # special case: dn is gotten in other field
        return _result[0][0]

    try:
        _ret = _result[0][1][property_str]
        if len(_ret) == 1: # only one result?
            return _ret[0]
    except KeyError:
        return '' # empty value

    return _ret

def get_groups(user):
```

(continues on next page)

(proviene de la página anterior)

```
# TODO only groups of organization or all of them?
_result = get_ldap_property('(cn=%s)' % user, 'groupMembership')
if not _result:
    return '' # no groups found

# only one result?
if type(_result) is str:
    _result = [_result]

_ret = ''
for _item in _result:
    _t = _item.split(',')
    if '=' in _t[0]:
        _ret = '%s%s, ' % (_ret, _t[0].split('=')[1])

return _ret[:-2] # remove trailing ',

def run():
    global global_ldap_object
    global_ldap_object = ldap.initialize('ldap://%s:389' % LDAP_SERVER)

    user=migasfree_client.utils.get_current_user().split("~")[0]
    print get_groups(user),

if __name__ == '__main__':
    run()
```

4.3.2 Sobre el cliente migasfree

El cliente migasfree devuelve el mensaje: «firma no válida»

Causa

Las claves almacenadas en el cliente no coinciden con el proyecto indicado en `/etc/migasfree.conf`.

Solución

Borra las claves del equipo cliente.

Para la versión de migasfree-client 4.6 ó inferior usa:

```
# rm /root/.migasfree-keys/*
```

Para la versión de migasfree-client 4.7 ó superior usa:

```
# rm -rf /var/migasfree-client/keys/[server]/*
```

Nota: Si es necesario, vuelve a registrar el cliente ejecutando: `migasfree --register`.

El cliente migasfree devuelve el mensaje: «Autoregistrando ordenador... Error: Error genérico»

Causa

Este error suele estar relacionado con el propietario de la carpeta `keys` del servidor.

Solución

Si has migrado el servidor desde una versión inferior a la 4.14, asigna al directorio `keys` y su contenido al propietario 890

```
# chown -R 890:890 /var/lib/migasfree/${FQDN}/keys
```

Imposible obtener /PKGS/binary-amd64/Packages 404 Not Found

Causa

Por defecto los repositorios físicos en el servidor se generan para la arquitectura i386.

Solución

Accede a Configuración - Sistemas de gestión de paquetes - `apt-get` y modifica el campo crear repositorio de esta manera:

```
cd %PATH%
mkdir -p %REPONAME%/PKGS/binary-i386/
mkdir -p %REPONAME%/PKGS/binary-amd64/
mkdir -p %REPONAME%/PKGS/sources/
cd ..
dpkg-scanpackages -m dists/%REPONAME%/PKGS /dev/null | gzip -9c > dists/%REPONAME%/
↪PKGS/binary-i386/Packages.gz
dpkg-scanpackages -m dists/%REPONAME%/PKGS /dev/null | gzip -9c > dists/%REPONAME%/
↪PKGS/binary-amd64/Packages.gz
dpkg-scansources dists/%REPONAME%/PKGS /dev/null | gzip -9c > dists/%REPONAME%/PKGS/
↪sources/Sources.gz
```

No se consigue ejecutar el cliente de migasfree en un cron.

Causa

Las variables de entorno no están disponibles.

Solución

En vez de programar el cron así:

```
00 07 * * * root /usr/bin/migasfree --update
```

debemos forzar el **entorno de usuario** de esta forma:

```
00 07 * * * root su -c 'migasfree --update' --login
```

4.4 Resolución de problemas

Un problema deja de serlo si no tiene solución.

—Eduardo Mendoza

A menudo, puede ocurrir que migasfree no esté funcionando como se espera. Para obtener más información y averiguar qué te puede estar ocurriendo, puedes poner tanto al cliente como al servidor en modo `DEBUG`.

4.4.1 Cliente en modo `DEBUG`

Simplemente debes poner el ajuste `Debug` a `True` en los *Ajustes del cliente migasfree*.

Cuando ejecutes `migasfree --update` en este modo desde una consola, verás en la salida estándar más información de la habitual.

También puede serte útil consultar la información que se va generando en `/var/tmp/migasfree.log`.

4.4.2 Servidor en modo `DEBUG`

Al estar el servidor realizado con Django, puedes usar el ajuste `DEBUG` a `True`. Este ajuste del servidor debes ponerlo en el fichero `/var/lib/migasfree/FQDN/conf/settings.py` y después reiniciar el servidor web.

Al hacer esto, la página del servidor migasfree que te está fallando te mostrará, en vez de una página de error escueta, otra página de error con información muy extensa y que te permitirá ver qué está ocurriendo.

Nota: Nunca dejes el modo `DEBUG` en un entorno de producción por motivos de seguridad.

4.4.3 Activar el sistema de logging en el servidor

El servidor utiliza el módulo de `logging` de Python.

Para activar el sistema de logging en el servidor de migasfree añade el ajuste `LOGGING` en `/var/lib/migasfree/FQDN/conf/settings.py`:

```
LOGGING = {
    'version': 1,
    'formatters': {
        'verbose': {
            'format': '%(levelname)s %(asctime)s %(module)s %(process)d
%(thread)d %(message)s'
        },
        'simple': {
            'format': '%(levelname)s %(message)s'
        },
    },
    'handlers': {
        'file': {
            'level': 'DEBUG',
            'class': 'logging.FileHandler',
            'filename': '/tmp/migasfree.log',
            'formatter': 'verbose',
        },
        'console': {
            'level': 'DEBUG',
            'class': 'logging.StreamHandler',
            'formatter': 'verbose',
        },
    },
    'loggers': {
        'migasfree': {
            'handlers': ['console', 'file'],
            'level': 'DEBUG',
```

(continues on next page)

(proviene de la página anterior)

```
}  
  }  
}
```

Esto hará que en el fichero indicado (/tmp/migasfree.log) se almacenen los logs.

La triangulación de los espárragos blancos.

5.1 Ajustes del servidor migasfree

No hay inteligencia allí donde no hay cambio ni necesidad de cambio.

—Herbert George Wells

Los ajustes de configuración del servidor migasfree se asignan en el fichero `/var/lib/migasfree/FQDN/conf/settings.py`.

Nota: Este es un fichero `python`, por lo que hay que llevar cuidado con la sintaxis y la indentación.

5.1.1 Ajustes propios de migasfree

MIGASFREE_ORGANIZATION

Valor por defecto: “My Organization”

Establece el nombre de tu organización.

Ejemplo:

```
MIGASFREE_ORGANIZATION = "ACME Corporation"
```

MIGASFREE_AUTOREGISTER

Valor por defecto: True

Especifica si los ordenadores pueden autoregistrar la plataforma y el proyecto al ejecutar `migasfree --update`.

Si no quieres que ningún ordenador registre proyectos y/o plataformas automáticamente, tienes que darlas de alta manualmente y asignar este ajuste a False.

Ejemplo:

```
MIGASFREE_AUTOREGISTER = False
```

MIGASFREE_DEFAULT_COMPUTER_STATUS

Valor por defecto: “intended”

Establece el estado por defecto que tendrá el ordenador cuando es añadido en el sistema.

Los valores que se pueden asignar son: “intended”, “reserved”, “unknown”, “in repair”, “available” o “unsubscribed”

Ejemplo:

```
MIGASFREE_DEFAULT_COMPUTER_STATUS = 'available' # Estado disponible por defecto
```

MIGASFREE_COMPUTER_SEARCH_FIELDS

Valor por defecto: (“id”, “name”,)

Establece los campos del modelo `Computer` por los que se podrá buscar un ordenador. El primer campo es importante ya que será el que aparezca en la primera columna de la lista de ordenadores. Si quieres ver el nombre del ordenador en vez del `id` en la lista de ordenadores, asigna el campo `name` el primero de la lista.

Se utiliza también para las búsquedas de un CID por los campos especificados. Ver *Fórmulas específicas*.

Ejemplo:

```
MIGASFREE_COMPUTER_SEARCH_FIELDS = ("name", "ip_address") # Búsquedas por nombre e IP
```

MIGASFREE_TMP_DIR

Valor por defecto: “/tmp”

Asigna la ruta donde se almacenarán los ficheros temporales generados por el servidor.

Ejemplo:

```
MIGASFREE_TMP_DIR = "/tmp/server"
```

MIGASFREE_PUBLIC_DIR

Valor por defecto: “/var/migasfree/repo”

Directorio donde se guardarán los paquetes y repositorios de físicos cada uno de los proyectos.

Ejemplo:

```
MIGASFREE_PUBLIC_DIR = "/var/repositories"
```

MIGASFREE_SECONDS_MESSAGE_ALERT

Valor por defecto: 1800

Si un ordenador tarda más de los segundos especificados en este ajuste en enviar un mensaje mientras se está actualizando, se considera que el ordenador va retrasado (Delayed). Normalmente, esto ocurre cuando se ha perdido la conexión con el servidor por cualquier circunstancia, por ejemplo cuando el usuario ha apagado el equipo antes de que el cliente termine el proceso de actualización. De esta forma, se queda registrado en el servidor como `Delayed`.

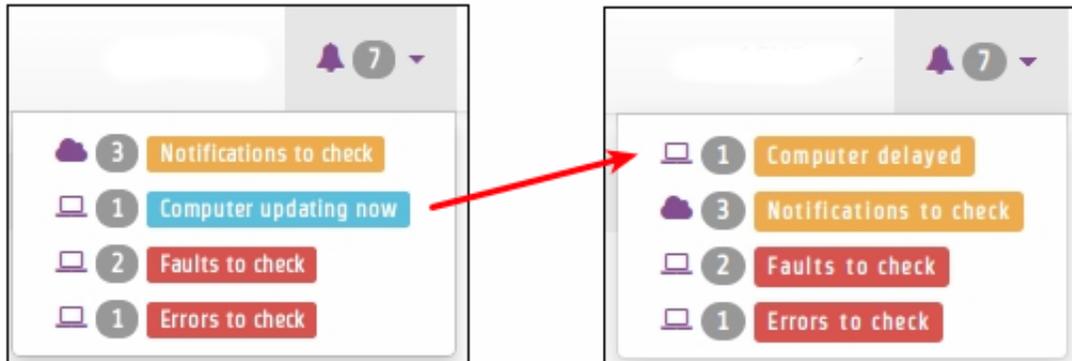


Figura 1: Un equipo pasando a retrasado.

Ejemplo:

```
MIGASFREE_SECONDS_MESSAGE_ALERT = 3600 # Una hora
```

MIGASFREE_HELP_DESK

Valor por defecto: “Put here how you want to be found”

Texto que aparece al ejecutar el comando del cliente `migasfree-label` para indicar al usuario cómo ponerse en contacto con Asistencia Técnica.

El comando `migasfree-label` tiene la finalidad de identificar inequívocamente al cliente. Este comando ejecutado en un cliente con entorno gráfico abrirá el navegador web mostrando una pequeña etiqueta que debe ser impresa y pegada en el ordenador con objeto de facilitar la asistencia técnica aún estando el ordenador apagado.

También es posible imprimir la etiqueta de un ordenador desde el servidor desplegando el menú contextual del ordenador y pulsando en `Etiqueta [ordenador]`.



Figura 2: Comando `migasfree-label`.

Ejemplo:

```
MIGASFREE_HELP_DESK = "Teléfono Asistencia Técnica: 555.12.34.56"
```

MIGASFREE_REMOTE_ADMIN_LINK

Valor por defecto: []

Cuando se asigna un valor a este ajuste, apaceren nuevas acciones por cada ordenador. El objetivo es poder ejecutar algún código desde nuestro equipo hacia el equipo que se quiere administrar. Generalmente se usa para acceder por vnc o ssh a los ordenadores, pero puede ser utilizado con cualquier otro fin.

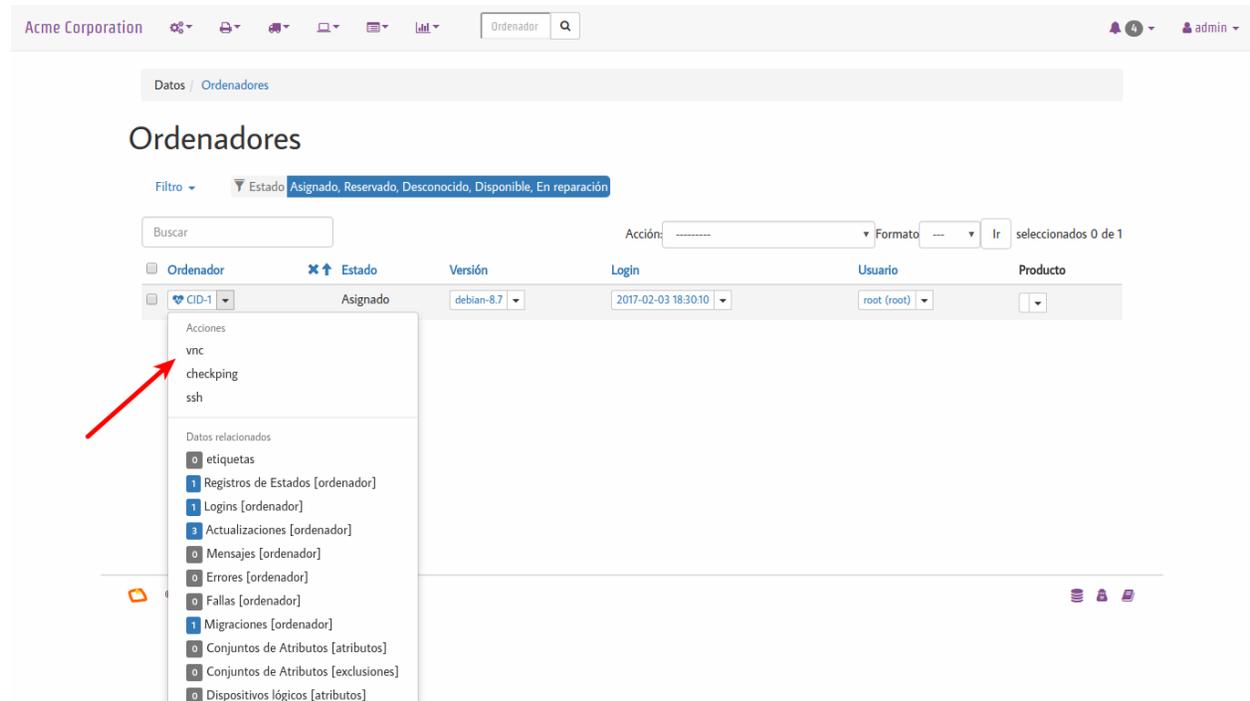


Figura 3: MIGASFREE_REMOTE_ADMIN_LINK

Las variables que se pueden usar dentro de este ajuste son:

{{ computer.<FIELD> }} para cualquier campo del modelo Computer

{{<<PROPERTYPREFIX>>}} cualquier propiedad del equipo cliente

Ejemplo vía ssh:

```
MIGASFREE_REMOTE_ADMIN_LINK = ["ssh://root@{{computer.ip_address}}"]
```

Ejemplo vía https y puerto (este último definido como propiedad PRT):

```
MIGASFREE_REMOTE_ADMIN_LINK = ["https://myserver/?computer={{computer.name}}&port={  
↔{PRT}}"]
```

Pueden usarse varios protocolos:

```
MIGASFREE_REMOTE_ADMIN_LINK = [  
  "vnc://{{computer.ip_address}}",  
  "checkping://{{computer.ip_address}}",  
  "ssh://root@{{computer.ip_address}}",  
]
```

Evidentemente, el navegador con el que se accede a la web del servidor debe saber cómo interpretar dichos protocolos. Por ejemplo, si usas Firefox y quieres permitir el protocolo `vnc` debes acceder a la dirección `about:config` y añadir:

```
network.protocol-handler.expose.vnc false
```

Luego crea un fichero ejecutable para asociarlo al protocolo `vnc` para que lance `vinagre` contra la IP del ordenador:

```
#!/bin/bash
URL=${1#vnc://}
vinagre $URL
```

MIGASFREE_HW_PERIOD

Valor por defecto: 30

Período en días para el envío del hardware de los ordenadores al servidor. Si han pasado más días de los especificados, se envía de nuevo toda la información del hardware al servidor.

Ejemplo:

```
MIGASFREE_HW_PERIOD = 1 # Cada día
```

MIGASFREE_INVALID_UUID

Valor por defecto =

```
[
  "03000200-0400-0500-0006-000700080008", # ASROCK
  "00000000-0000-0000-0000-000000000000",
  "FFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFFFF",
  "00000000-0000-0000-0000-FFFFFFFFFFFFFF",
]
```

Es una lista con UUIDs inválidos.

Algunas placas base pueden no tener asignado un UUID único, pudiéndose darse el caso que varios equipos tengan un mismo UUID. Esto provoca que en el servidor se comparta el mismo registro de ordenador.

Para evitarlo, es preciso añadir estos UUIDs en este ajuste.

Cuando un ordenador tiene un UUID inválido, el servidor toma y asigna el nombre del ordenador como UUID.

Puedes hacer UUIDs inválidos añadiendo en `/var/lib/migasfree/FQDN/conf/settings.py` la siguiente instrucción:

```
MIGASFREE_INVALID_UUID.extend( my_invalid_UUIDs )
```

donde `my_invalid_UUIDs` es una lista de UUIDs invalidos.

Ejemplo:

```
MIGASFREE_INVALID_UUID.extend( ["00000000-FFFF-FFFF-FFFF-FFFFFFFFFFFFFF",] )
```

MIGASFREE_NOTIFY_NEW_COMPUTER

Valor por defecto = False

Si se asigna a `True`, el sistema añadirá una Notificación cuando un cliente migasfree se registra en el servidor por primera vez.

MIGASFREE_NOTIFY_CHANGE_UUID

Valor por defecto = `False`

Si se establece a `True` se creará una Notificación cuando un equipo cambia de UUID.

Esto puede ocurrir en contadas ocasiones y está relacionado con antiguos clientes de migasfree, UUIDs inválidos, o con cambios de placa base en el ordenador.

MIGASFREE_NOTIFY_CHANGE_NAME

Valor por defecto = `False`

Si se establece a `True`, se creará una Notificación cuando se detecta que un ordenador ha cambiado de nombre.

Este ajuste puede resultar útil para detectar UUIDs no únicos.

MIGASFREE_NOTIFY_CHANGE_IP

Valor por defecto = `False`

Si se establece a `True` se creará una Notificación cuando un ordenador cambia de ip.

Este ajuste puede resultar útil para detectar UUIDs no únicos.

Nota: No actives este ajuste si tienes ordenadores con IP dinámica, ya que se crearán demasiadas notificaciones irrelevantes.

5.1.2 Ajustes de Django

Los ajustes de Django también pueden ser modificados para adaptar el funcionamiento del servidor añadiendo el ajuste en el fichero `/var/lib/migasfree/FQDN/conf/settings.py`.

El más importante de este tipo de ajustes es:

DATABASES

Valor por defecto:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'migasfree',
        'USER': 'migasfree',
        'PASSWORD': 'migasfree',
        'HOST': '',
        'PORT': '',
    }
}
```

5.2 Ajustes del cliente migasfree

Uno no debe adaptarse al cambio, sino crearlo.

—Jorge González Moore

Los ajustes de configuración del cliente migasfree se encuentran en el fichero `/etc/migasfree.conf`

5.2.1 Sección [client]

Server

Valor por defecto: `localhost`

Nombre del servidor migasfree contra el que se van a realizar las actualizaciones.

Ejemplo:

```
Server = 192.168.1.10
```

Project

Valor por defecto: Se basa en la función de python `platform.linux_distribution()`

Nombre del proyecto migasfree. Sería el equivalente al nombre que le quieras dar a tu *distribución personalizada*. Es muy *recomendable* que configures este ajuste, ya que para algunas distribuciones la función `platform.linux_distribution()` puede producir versiones diferentes al aumentar de release (CentOS sería un ejemplo de esto).

Ejemplo:

```
Project = MiDistro-1
```

Computer_Name

Valor por defecto: Se obtiene de la función de python `platform.node()`

Nombre del ordenador que se mostrará en migasfree. Si por cualquier circunstancia se necesita que el nombre del ordenador no sea el `hostname`, puedes configurar este ajuste para modificarlo.

Ejemplo:

```
Computer_Name = PC15403
```

Debug

Valor por defecto: `False`

Si se establece a `True`, la salida de los comandos del cliente mostrará información útil para la depuración.

Ejemplo:

```
Debug = True
```

GUI_Verbose

Valor por defecto: True

Indica si aparecen más o menos mensajes en el Intefaz Gráfico de Usuario. Si se asigna a `False`, sólo se mostrarán el primer y último mensaje.

Ejemplo:

```
GUI_Verbose = False
```

Auto_Update_Packages

Valor por defecto: True

Determina si al ejecutar `migasfree --update` se instalan las nuevas versiones de los paquetes ya instalados.

Si se establece este ajuste a **False** las actualizaciones de paquetes no se producirán al ejecutar `migasfree` con objeto de que sea el usuario quien decida cuándo quiere realizarlas (siguiendo p.e. la configuración de un gestor de actualizaciones tipo `update-manager` de Gnome figura 17.1 o ejecutando una actualización desde el *front-end* del sistema de paquetería).

Este ajuste no afectará en ningún caso a los paquetes a instalar y/o a los paquetes a desinstalar que hubiera definidos en los repositorios de `migasfree`.

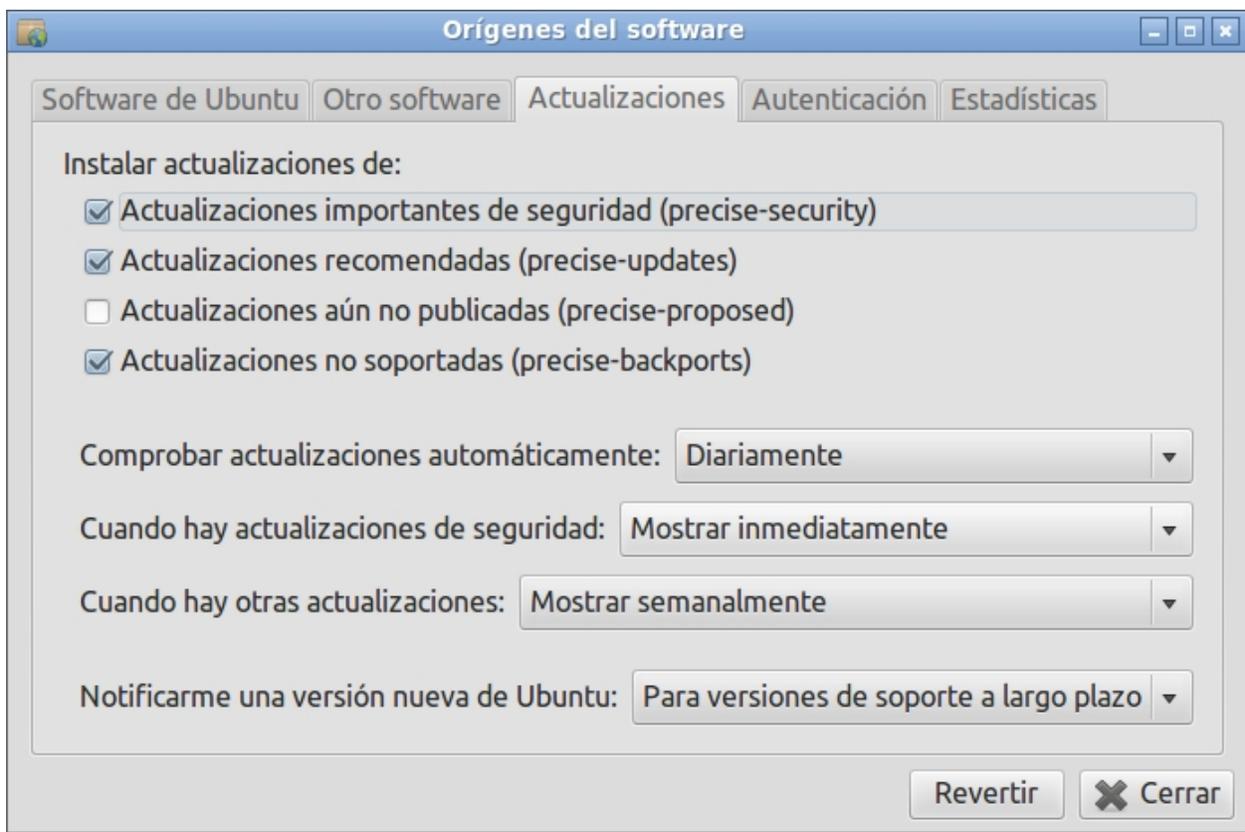


Figura 4: Configuración del Gestor de Actualizaciones.

Ejemplo:

```
Auto_Update_Packages = False
```

SSL_Cert

Valor por defecto: No establecido.

Ruta al fichero de certificado SSL de servidor en el cliente.

Si este fichero de certificado no existe se utilizará igualmente **https** para la privacidad, pero la autenticación entre extremos no estará garantizada. En este caso aparece en consola el siguiente mensaje:

```
Certificate does not exist and authentication is not guaranteed
```

Ejemplo:

```
SSL_Cert = "/path/to/ssl/cert"
```

Proxy

Valor por defecto: No establecido.

Configuración del proxy.

Ejemplo:

```
Proxy = 192.168.1.100:8080
```

Package_Proxy_Cache

Valor por defecto: No establecido.

Permite especificar la dirección de un sistema caché de repositorios como podría ser `apt-cacher`.

Ejemplo:

```
Package_Proxy_Cache = 192.168.1.101:1234
```

Consulta el apartado `Servicio de caché de paquetes` del capítulo *Migasfree en producción*.

5.2.2 Sección [packager]

Esta sección se utiliza cuando se suben paquetes al servidor mediante el comando `migasfree-upload`. Se te pedirá la información que no hayas especificado en estos ajustes.

User

Valor por defecto: No establecido.

Usuario con permisos para subir paquetes al servidor `migasfree`. Por defecto, la base de datos del servidor `migasfree` incluye el usuario `packager` con los permisos adecuados para almacenar paquetes en el servidor.

Ejemplo:

```
User = packager
```

Password

Valor por defecto: No establecido.

Contraseña del usuario.

Ejemplo:

```
Password = packager
```

Project

Valor por defecto: No establecido.

Indica el nombre del proyecto migasfree al que se van a subir los paquetes.

Ejemplo:

```
Project = AZLinux-12
```

Store

Valor por defecto: No establecido.

Almacén en el servidor migasfree donde se guardarán los paquetes. Corresponde al nombre de una carpeta en el servidor donde se situará el Paquete o Conjunto de Paquetes. Puedes ver la lista de almacenes disponibles accediendo a [Liberación - Almacenes](#) en la web del servidor migasfree. Si asignas un Almacén que no existe se creará automáticamente al subir el primer paquete.

Ejemplo:

```
Store = Acme # Sitúa en /var/migasfree/repo/<Project>/STORES/Acme los paquetes.
```

5.2.3 Variables de entorno

Mediante el uso de variables de entorno podemos modificar también la configuración del cliente migasfree.

MIGASFREE_CONF

Por defecto, el fichero de configuración del cliente migasfree se encuentra en `/etc/migasfree.conf` pero mediante la variable de entorno `MIGASFREE_CONF` podemos indicar al cliente que use otro fichero. Esto puede serte útil si tienes que subir paquetes mediante el comando `“migasfree-upload“` a distintos servidores migasfree desde la consola.

Ejemplo:

```
export MIGASFREE_CONF='/etc/migasfree.conf.serverA'  
migasfree-upload -f <mipaquete>
```

Además, todos los ajustes del fichero de configuración del cliente migasfree también pueden ser asignados mediante variables de entorno, siendo estas variables prioritarias frente a los ajustes del fichero de configuración:

MIGASFREE_CLIENT_SERVER

```
MIGASFREE_CLIENT_PROJECT
MIGASFREE_CLIENT_COMPUTER_NAME
MIGASFREE_CLIENT_DEBUG
MIGASFREE_CLIENT_GUI_VERBOSE
MIGASFREE_CLIENT_AUTO_UPDATE_PACKAGES
MIGASFREE_PROXY
MIGASFREE_CLIENT_PACKAGE_PROXY_CACHE
MIGASFREE_PACKAGER_USER
MIGASFREE_PACKAGER_PASSWORD
MIGASFREE_PACKAGER_PROJECT
MIGASFREE_PACKAGER_STORE
```

Como ejemplo de uso de las variables de entorno, imagina un escenario en el cual tienes un servidor migasfree y muchos centros en los que en cada uno de ellos hay un servicio de caché de paquetes para minimizar el tráfico de Internet. Para configurar cada equipo, deberías tener un paquete de configuración del cliente migasfree por cada centro, pero si tienes muchos centros esto puede resultar costoso. Una solución podría ser tener un sólo paquete de configuración del cliente migasfree para todos los centros y, en la postinstalación del paquete, crear las variables de entorno necesarias en función de la etiqueta del centro.

```
# Codigo de ejemplo postinst acme-migasfree-client

TAGS=`migasfree-tags -g`
for CENTRO in $TAGS
do
    if [ $CENTRO = "CTR-DELEGACION-BARCELONA" ]; then
        echo "MIGASFREE_CLIENT_PACKAGE_PROXY_CACHE='192.168.96.6:3142'" > /etc/profile.d/
↪migasfree.sh
    fi
    if [ $CENTRO = "CTR-DELEGACION-MADRID" ]; then
        echo ""MIGASFREE_CLIENT_PACKAGE_PROXY_CACHE='192.168.80.4:3142'" > /etc/profile.d/
↪migasfree.sh
    fi
done
```


La aproximación de la prestidigitación.

6.1 Empaquetando migasfree

Nadie es como otro. Ni mejor ni peor. Es otro. Y si dos están de acuerdo, es por un malentendido.

—Jean-Paul Sartre

El proceso consiste, básicamente, en bajarte el fuente del proyecto y ejecutar el comando `bin/create-package`

Nota: Si quieres la versión de desarrollo puedes bajar `master.zip` en vez de `latest.zip`

6.1.1 Creación del paquete migasfree-server (.deb)

Abre una terminal como **root** y baja el código fuente de migasfree:

```
wget https://github.com/migasfree/migasfree/archive/latest.zip
```

Necesitaremos tener instalado `unzip` para descomprimir el fichero `zip`:

```
apt-get install unzip
```

Descomprimimos el fichero `latest.zip`:

```
unzip latest.zip  
rm latest.zip
```

Ahora tendrás una carpeta llamada `migasfree-latest`.

Creamos a continuación el paquete `migasfree-server`. Para ello necesitamos tener instalado el paquete `python-stdeb`:

```
apt-get install python-stdeb
```

Nos situamos en la carpeta `bin` del proyecto y ejecutamos el script `create-package`:

```
cd migasfree-latest/bin
./create-package
cd ../../
```

Ahora en la carpeta `deb_dist` tenemos el paquete `deb` que instalamos:

```
dpkg -i migasfree-latest/deb_dist/migasfree-server*_all.deb
```

Por problemas de dependencias, seguramente se dejará sin configurar el servidor de `migasfree`. Para instalar las dependencias que faltan, haremos:

```
apt-get -f install
```

6.1.2 Creación del paquete `migasfree-client (.deb)`

Abre una terminal como **root** y baja el código fuente del cliente `migasfree`:

```
wget https://github.com/migasfree/migasfree-client/archive/latest.zip
```

Necesitaremos tener instalado `unzip` para descomprimir el fichero `zip`:

```
apt-get install unzip
```

Descomprimos el fichero `latest.zip`:

```
unzip latest.zip
rm latest.zip
```

Ahora tendrás una carpeta llamada `migasfree-client-latest`.

Creamos a continuación el paquete `migasfree-client`. Para ello necesitamos tener instalado el paquete `python-stdeb`:

```
apt-get install python-stdeb
```

Nos situamos en la carpeta `bin` del proyecto y ejecutamos el script `create-package`:

```
cd migasfree-client-latest/bin
./create-package
cd ../../
```

Ahora en la carpeta `deb_dist` tenemos el paquete `deb` que instalamos:

```
dpkg -i migasfree-client-latest/deb_dist/migasfree-client*_all.deb
```

Por problemas de dependencias, seguramente se dejará sin configurar el cliente de `migasfree`. Para instalar las dependencias que faltan, haremos:

```
apt-get -f install
```

6.1.3 Otras Distribuciones a las implementadas

Si al ejecutar `./create-package` te aparece:

```
Error: Computer distro is not available. Aborting package creation.
```

entonces, consigue el nombre de tu Distribución:

```
_DISTRO=$(python -c "import platform; print platform.linux_distribution()[0].strip()")
```

y crea un nuevo fichero cuyo nombre sea `$_DISTRO` en el directorio `setup.cfg.d` para paquetería rpm:

```
touch ../setup.cfg.d/$_DISTRO #
```

o en el directorio `stdeb.cfg.d` para paquetería deb:

```
touch ../stdeb.cfg.d/$_DISTRO
```

Finalmente, copia dentro de este fichero el contenido de otro fichero de una Distribución similar y modifica las dependencias necesarias.

Una vez realizado este proceso vuelve a ejecutar `./create-package`

6.2 Empaquetando proyectos python

La diferencia entre el pasado, el presente y el futuro es sólo una ilusión persistente.

—Albert Einstein

Es muy posible que en la distribución en la que has instalado el servidor migasfree no encuentres los paquetes que se requieren para instalar el servidor, bien porque simplemente no están disponibles o bien porque la versión disponible no es suficientemente reciente como para ejecutar el servidor.

En este capítulo vas a empaquetar `django` como ejemplo y el método que usarás es válido para la mayoría de proyectos escritos en `python`.

6.2.1 Creación del paquete `django` en distros basadas en paquetería `apt`

Descarga el código del proyecto `django` cuya versión necesites y descomprímelo:

```
wget https://www.djangoproject.com/download/1.6.2/tarball/ -O Django-1.6.2.tar.gz
tar xzvf Django-1.6.2.tar.gz
cd Django-1.6.2
```

Asegúrate que tienes instalado el paquete `python-stdeb`:

```
apt-get install python-stdeb
```

Ahora crea el paquete:

```
python setup.py --command-packages=stdeb.command bdist_deb
```

En la carpeta `deb_dist` tendrás el paquete `deb`.

Más información en <https://wiki.debian.org/Python/Packaging>

6.2.2 Creación del paquete django en distros basadas en paquetería rpm

Descarga el código del proyecto django cuya versión necesites y descomprímelo:

```
wget https://www.djangoproject.com/download/1.6.2/tarball/ -O Django-1.6.2.tar.gz
tar xzvf Django-1.6.2.tar.gz
cd Django-1.6.2
```

Ahora crea el paquete:

```
python setup.py bdist_rpm
```

En la carpeta `dist` tendrás el paquete rpm.

Para más información puedes consultar <http://docs.python.org/2.0/dist/creating-rpms.html>

El rompecabezas Vartabedian.

7.1 Bibliografía

Software Configuration Management, Bersoff, Henderson & Siegel, Prentice-Hall, 1980

[IEEE828-98] IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans, IEEE, 1998.

Eduardo Romero Moreno, [Migración Escritorio Software Libre](#), 2011

7.2 Referencias

[ConfigPackages](#)

[New debian policy](#)

[rpm.org](#)

[Metrica v3](#)

7.3 Glosario de términos

Auditoría. Análisis cuyo objetivo es revisar y evaluar la gestión efectuada.

Alerta. Aviso para que el usuario preste su atención a una determinada situación.

Atributo. Valor concreto que toma una fórmula al ser ejecutada en un equipo.

Cambio. Actividad que modifica un ECS, generando una nueva versión del ECS.

CID. *Computer Identifier*. Número que identifica inequívocamente a una placa base y, por tanto, a un ordenador.

ECS. Elemento de configuración software. Objeto software sometido a la GCS.

Falla. Hecho negativo que se produce en un equipo cliente.

GCS. Gestión de la configuración software. Proceso de la Ingeniería del Software que identifica, hace seguimiento y controla cada uno de los cambios que se producen en los sistemas.

Liberación. Actividad de situar una versión del ECS en un despliegue para que los clientes del ECS puedan acceder a él e instalarlo.

Metadatos. Información sobre los datos.

Paquete. Contenedor que encapsula un conjunto de ECS liberados junto a sus metadatos.

Paquete huérfano. Paquete que no está en ningún despliegue.

PMS. Package Management System - Gestor de paquetes. Programa que permite aplicar en un equipo, los cambios que han sido liberados en los despliegues.

Fórmula. Código que devolverá una determinada característica de los equipos o de los usuarios.

Repositorio. Almacén de ECS.

Despliegue migasfree. Añade a un repositorio la capacidad de establecer cuándo y quién tiene acceso al repositorio.

Systems Management System. Gestor de sistemas. Software que permite la administración de escritorios de manera centralizada.

7.4 API

La belleza es el acuerdo entre el contenido y la forma.

—Henrik Johan Ibsen

El objetivo de este capítulo es describir la API de migasfree.

Desde sus inicios migasfree ha contado con una API propia pero desde la versión 4.11 del servidor se está incorporado una nueva API REST al servidor migasfree con objeto de sustituirla totalmente.

Puedes consultar la autodocumentación de la API REST de migasfree accediendo a <http://<miservidor>/docs> o haciendo click en el icono del espía que aparece situado abajo a la derecha en todas las páginas de la aplicación web del servidor.

El proyecto `fun-with-migasfree-examples` incluye un directorio llamado `api-examples` donde encontrarás código de ejemplo para hacer uso de la API REST.

Si aún no te has bajado los ejemplos ahora puede ser un buen momento para hacerlo y echarle un ojo al uso de la API REST.

```
$ wget https://github.com/migasfree/fun-with-migasfree-examples/archive/master.zip
$ unzip master.zip
$ cd fun-with-migasfree-examples-master/api-examples
```

Es necesario, para usar estos ejemplos, que tengas instalado el `migasfree-sdk`.

Puedes ejecutar esta instrucción para instalarlo:

```
# wget -O - http://migasfree.org/pub/install-sdk | bash
```

En cuanto a la API antigua (y que va a desaparecer en el futuro) podemos decir que tiene 2 tipos de API:

- Pública:

- Son accesibles públicamente.
- Utilizan el metodo GET de HTTP para el envío de parámetros

■ Privada:

- Comparten entre sí la misma manera de llamar a las funciones.
- Envían un fichero en el parámetro message del método POST HTTP, cuyo nombre de fichero debe usar el siguiente formato:

```
<COMPUTER_NAME>.<UUID>.<API_FUNCTION>
```

- En el contenido del fichero message se introduce, en formato json lo que denominamos el API Private Data con los parámetros de entrada de las funciones.
- En la mayoría de los casos este API Private Data es firmado usando la KEY del Proyecto o del Empaquetador añadiendo esta firma al final del fichero:

```
``API Private Data`` + sign(``API Private Data``)
```

- En el menor de los casos estas funciones en vez de firma requieren el uso de un usuario y contraseña que simplemente se introduce en el propio API Private Data.
- Los valores devueltos por la funciones se reciben en formato json convenientemente firmados por el servidor, siguiendo la estructura:

```
{"<API_FUNCTION>.return": ``API Private Data`` } +  
  sign({"<API_FUNCTION>.return": ``API Private Data`` })
```

7.4.1 get_projects

Pertenece a la API pública.

Devuelve una lista de diccionarios de las plataformas con sus `projects` correspondientes.

Parámetros de entrada

- Ninguno

Salida

- Lista de diccionarios de plataformas:
 - **platform**: Nombre de la plataforma.
 - **projects**: Lista de diccionarios de proyectos:
 - **name**: Nombre del proyecto.

Veamos un ejemplo. Si en un navegador web accedemos a la siguiente dirección:

```
http://miservidor/get_projects
```

Obtendremos una cadena de texto en formato json parecida a esta:

```
[{"platform": "Linux", "projects": [{"name": "ACME-1"}, {"name": "debian-7.0"}]}
```

7.4.2 get_computer_info

Pertenece a la API pública.

Obtiene un diccionario con información relevante del ordenador consultado.

Parámetros de entrada

- **uuid**: Como parámetro de método GET en la petición HTTP debe indicarse el identificador único de la placa base del ordenador.

Salida

- Diccionario de datos del equipo:
 - **search**: Valor del primer campo indicado en la lista `MIGASFREE_COMPUTER_SEARCH_FIELDS` de los *Ajustes del servidor migasfree* y que sirve para facilitar la identificación del equipo en vez de usar el `uuid`.
 - **name**: Nombre del ordenador.
 - **tags**: Lista de cadenas de texto con los nombres de las Etiquetas asignadas al ordenador.
 - **available_tags**: Diccionario de Propiedades de tipo Etiqueta.
 - **<Propiedad>**: Lista de cadenas de texto con el nombre de las etiquetas.
 - **helpdesk**: Cadena de texto `MIGASFREE_HELP_DESK` de los *Ajustes del servidor migasfree*
 - **id**: Número identificador del ordenador en la tabla de Ordenadores
 - **uuid**: Identificador único de la placa base del ordenador

Veamos un ejemplo accediendo a:

```
http://miservidor/get_computer_info/?uuid=E9E66900-CBD4-9A47-B2EC-6ED0367A3AFB
```

obtendríamos algo parecido a esto:

```
{"search": 2, "name": "debian-client", "tags": [], "available_tags": {},  
"helpdesk": "Put here how you want to be found", "id": 2,  
"uuid": "E9E66900-CBD4-9A47-B2EC-6ED0367A3AFB"}
```

7.4.3 computer_label

Pertenece a la API pública.

Obtiene una página html que muestra la etiqueta que debe pegarse físicamente en el equipo para facilitar su identificación aún estando éste apagado.

Es utilizada por el comando `migasfree-label`.

Parámetros de entrada

- **uuid**: Como parámetro de método GET en la petición HTTP debe indicarse el identificador único de la placa base del ordenador.

Salida

- La página **html** de la etiqueta:

Por ejemplo al ejecutar:

```
http://miservidor/computer_label/?uuid=E9E66900-CBD4-9A47-B2EC-6ED0367A3AFB
```

Podemos obtener algo como:

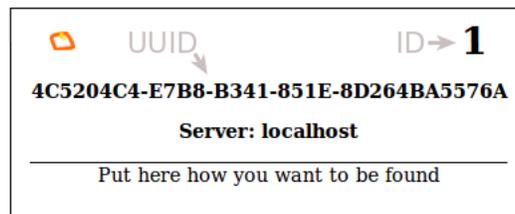


Figura 1: Comando `migasfree-label`.

7.4.4 register_computer

Pertenece a la API Privada.

Necesita usuario y contraseña con permisos de lectura/escritura en Ordenadores, y en Plataformas y Proyectos si MIGASFREE_AUTOREGISTER está activo. Ver *Ajustes del servidor migasfree*

Esta función realiza lo siguiente:

- Registra el Ordenador en el servidor.
- Añade la Plataforma y/o Proyecto del ordenador si no existen, siempre y cuando MIGASFREE_AUTOREGISTER esté activo.
- Añade las correspondientes Notificaciones
- Como resultado se obtendrán las KEYS del Proyecto que usarán las funciones de la API pública que las requieren.

API Private Data Input

- **username:** Nombre del usuario
- **password:** Contraseña
- **platform:** Plataforma del ordenador.
- **project:** Proyecto del ordenador.
- **pms:** Sistema de paquetería.
- **ip:** Dirección IP.

API Private Data Output

- **migasfree-server.pub:** KEY pública del servidor,
- **migasfree-client.pri:** KEY privada del proyecto.
- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.4.5 get_key_packager

Pertenece a la API Privada.

Necesita usuario y contraseña con permisos de lectura/escritura en Paquetes.

Obtiene la KEY que permitirá subir paquetes al servidor con el comando de cliente `migasfree-upload`.

API Private Data Input

- **username:** Nombre del usuario.
- **password:** Contraseña del usuario.

API Private Data Output

- **migasfree-server.pub:** KEY pública del servidor,
- **migasfree-packager.pri:** KEY privada del empaquetador.
- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.4.6 upload_server_package

Pertenece a la API Privada.

Necesita firmar con KEY de empaquetador.

Sube un paquete al servidor.

El fichero del paquete debe enviarse en `HttpRequest.FILES["package"]`

API Private Data Input

- **project:** Proyecto.
- **store:** Almacén donde se almacena el paquete.
- **source:** Valor booleano que indica si el paquete es el binario o el fuente.

API Private Data Output

- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.4.7 upload_server_set

Pertenece a la API Privada.

Necesita firmar con KEY de Empaquetador.

Sube un paquete de un Conjunto de Paquetes al servidor .

El fichero del paquete debe enviarse en `HttpRequest.FILES["package"]`

API Private Data Input

- **project:** Proyecto.
- **store:** Almacén donde se almacena el paquete.
- **packageset:** Conjunto de Paquetes en el que está incluido el paquete.

API Private Data Output

- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.4.8 create_repositories_of_packageset

Pertenece a la API Privada.

Necesita firmar con KEY de Empaquetador.

Se utiliza para forzar la creación de los metadatos de los Despliegues en donde está asignado el Conjunto de Paquetes especificado.

Se usa despues de subir todos los paquetes de un Conjunto de Paquetes.

API Private Data Input

- **packageset:** El nombre del Conjunto de Paquetes.
- **project:** El Proyecto del Conjunto de Paquetes.

API Private Data Output

- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.4.9 upload_computer_message

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Envia un mensaje de texto al servidor informando que proceso esta realizando el cliente. Es utilizado por migasfree

--update

API Private Data Input

El mensaje de texto que se quiere enviar al servidor.

API Private Data Output

- **errmfs**: Diccionario con el posible error devuelto.
 - **code**: Código del error. Un valor de cero indica que no ha habido error.
 - **info**: Texto descriptivo del error.

7.4.10 get_properties

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Obtiene las Propiedades activas en el servidor migasfree.

API Private Data Input

No requiere.

API Private Data Output

- **properties:** Lista de diccionarios con las Propiedades:
 - **prefix:** Prefijo de la propiedad
 - **function** Instrucciones de la Propiedad
 - **language** Lenguaje en que está programado la propiedad.
- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.4.11 upload_computer_info

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Dados los datos del ordenador Obtiene del servidor diferente información con lo que el cliente debe hacer para realizar una actualización.

API Private Data Input

- **computer:** Diccionario con información relativa al Ordenador
 - **hostname:** Nombre del ordenador.
 - **ip:** Dirección ip del ordenador.
 - **platform:** Plataforma.
 - **project:** Nombre del proyecto.
 - **user:** Cuenta del usuario que esta logueado en la sesión gráfica.
 - **user_fullname:** Nombre completo del usuario
- **attributes:** Lista de diccionarios con los Atributos conseguidos al ejecutar cada una de las Propiedades
 - **<ATTRIBUTES_NAME>:** Valor del Atributo

API Private Data Output

- **faultsdef:** Lista de diccionarios de Definiciones de Fallas
 - **name:** Nombre de la falla.
 - **function:** Instrucciones de la falla.
 - **language:** Lenguaje en que está escrita la falla.
- **repositories:** Lista de diccionarios de repositorios que deben configurarse en el cliente y que han sido seleccionados por el servidor en función de los atributos de entrada y la fecha actual.
 - **name:**
- **packages:** Diccionario de paquetes.
 - **install:** Lista de cadenas de texto con los paquetes a instalar.
 - **remove:** Lista de cadenas de texto con los paquetes a desinstalar.
 - **hardware_capture:** True si el ordenador tiene que enviar el hardware.
 - **devices:** #TODO
- **logical:** Diccionario de dispositivos a instalar o desinstalar.
- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.4.12 upload_computer_faults

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Sube el resultado de las Fallas.

API Private Data Input

Diccionario con las Fallas:

- **<PROPIEDAD>**: Texto de la salida estándar al ejecutar la FALLA

API Private Data Output

- **errmfs**: Diccionario con el posible error devuelto.
 - **code**: Código del error. Un valor de cero indica que no ha habido error.
 - **info**: Texto descriptivo del error.

7.4.13 upload_computer_hardware

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Sube el hardware del Ordenador.

API Private Data Input

Salida en formato *json* del comando `lshw`.

API Private Data Output

- **errmfs**: Diccionario con el posible error devuelto.
 - **code**: Código del error. Un valor de cero indica que no ha habido error.
 - **info**: Texto descriptivo del error.

7.4.14 upload_computer_software_base_diff

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Sube la diferencia respecto al ordenador base

API Private Data Input

Texto con la lista de paquetes respecto al ordenador base separados por retornos de carro.

API Private Data Output

- **errmfs**: Diccionario con el posible error devuelto.
 - **code**: Código del error. Un valor de cero indica que no ha habido error.
 - **info**: Texto descriptivo del error.

7.4.15 upload_computer_software_base

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Lo utiliza el Ordenador de referencia para informar de los paquetes que tiene instalados.

API Private Data Input

Texto con la lista de paquetes instalados separados por retornos de carro.

API Private Data Output

- **errmfs**: Diccionario con el posible error devuelto.
 - **code**: Código del error. Un valor de cero indica que no ha habido error.
 - **info**: Texto descriptivo del error.

7.4.16 upload_computer_software_history

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Informa de cambio en el software.

API Private Data Input

Texto con el cambio de paquetes producidos en el Ordenador. Sigue el formato:

```
# [<FECHA DESDE>, <FECHA_HASTA>
<ACTION><PACKAGE> ,
<ACTION><PACKAGE> , ...
```

dónde ACTION puede ser (-) para indicar desinstalado y (+) para indicar paquete instalado.

API Private Data Output

- **errmfs**: Diccionario con el posible error devuelto.
 - **code**: Código del error. Un valor de cero indica que no ha habido error.
 - **info**: Texto descriptivo del error.

7.4.17 `get_computer_software`

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Obtiene el conjunto de paquetes del `Ordenador` de referencia.

API Private Data Input

No requiere.

API Private Data Output

- Texto con la lista de paquetes del `Ordenador` de referencia separados por retorno de carro
- **errmfs**: Diccionario con el posible error devuelto.
 - **code**: Código del error. Un valor de cero indica que no ha habido error.
 - **info**: Texto descriptivo del error.

7.4.18 upload_computer_errors

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Sube los errores producidos en el cliente.

API Private Data Input

Texto con el errores que han producido en el cliente.

API Private Data Output

- **errmfs**: Diccionario con el posible error devuelto.
 - **code**: Código del error. Un valor de cero indica que no ha habido error.
 - **info**: Texto descriptivo del error.

7.4.19 get_computer_tags

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Obtiene las etiquetas del Ordenador y las disponibles en el sistema.

API Private Data Input

No requerido

API Private Data Output

- **selected:** Lista de textos con las Etiquetas asignadas al ordenador.
- **available:** Diccionario de etiquetas.
 - **<PROPERTY>**: Lista de textos con las Etiquetas disponibles por cada Propiedad` de tipo ``tag
- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.4.20 set_computer_tags

Pertenece a la API Privada.

Necesita firmar con KEY de Proyecto.

Asigna las etiquetas al ordenador y como resultado se obtiene los paquetes que deben instalarse y desinstalarse en función de las etiquetas que anteriormente tuviera asignadas el equipo.

API Private Data Input

- **tags:** Lista de etiquetas a asignar al Ordenador

API Private Data Output

- **packages:** Diccionario con la listas de paquetes.
 - **preinstall:** Lista de nombres de paquetes separados por espacios obtenidos del campo `default preinstall packages`
 - **install:** Lista de nombres de paquetes separados por espacios obtenidos del campo `default install packages`
 - **remove:** Lista de nombres de paquetes separados por espacios obtenidos del campo `default remove packages`
- **errmfs:** Diccionario con el posible error devuelto.
 - **code:** Código del error. Un valor de cero indica que no ha habido error.
 - **info:** Texto descriptivo del error.

7.5 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document «free» in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of «copyleft», which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The «Document», below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as «you». You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A «Modified Version» of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A «Secondary Section» is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The «Invariant Sections» are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The «Cover Texts» are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A «Transparent» copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not «Transparent» is called «Opaque».

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The «Title Page» means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, «Title Page» means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The «publisher» means any person or entity that distributes copies of the Document to the public.

A section «Entitled XYZ» means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as «Acknowledgements», «Dedications», «Endorsements», or «History».) To «Preserve the Title» of such a section when you modify the Document means that it remains a section «Entitled XYZ» according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

1. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
2. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
3. State on the Title page the name of the publisher of the Modified Version, as the publisher.
4. Preserve all the copyright notices of the Document.
5. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
6. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
7. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
8. Include an unaltered copy of this License.

9. Preserve the section Entitled «History», Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled «History» in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
10. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the «History» section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
11. For any section Entitled «Acknowledgements» or «Dedications», Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
12. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
13. Delete any section Entitled «Endorsements». Such a section may not be included in the Modified Version.
14. Do not retitle any existing section to be Entitled «Endorsements» or to conflict in title with any Invariant Section.
15. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled «Endorsements», provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled «History» in the various original documents, forming one section Entitled «History»; likewise combine any sections Entitled «Acknowledgements», and any sections Entitled «Dedications». You must delete all sections Entitled «Endorsements».

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an «aggregate» if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled «Acknowledgements», «Dedications», or «History», the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License «or any later version» applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free

Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

«Massive Multiauthor Collaboration Site» (or «MMC Site») means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A «Massive Multiauthor Collaboration» (or «MMC») contained in the site means any set of copyrightable works thus published on the MMC site.

«CC-BY-SA» means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

«Incorporate» means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is «eligible for relicensing» if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled «GNU Free Documentation License».

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the «with. . . Texts.» line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.