

Gestión de servicios para la empresa

Business Service Management

Remo Suppi Boldrito

PID_00241834

Índice

Introducción	5
1. Sistema de gestión de contenidos	7
1.1. Drupal	7
1.2. WordPress.....	9
1.3. Moodle	11
2. Monitorización	14
2.1. Graphite	14
2.2. OpenNMS.....	16
2.3. Zenoss (Zenoss Core).....	17
3. Gestión de proyectos: Redmine	19
4. Cloud Storage	20
5. Sistemas operativos virtuales o Cloud OS	22
6. Sitios de colaboración	24
7. ERP	26
8. Otras aplicaciones útiles en el <i>back office</i>	29
Actividades	32
Bibliografía	33

Introducción

Existen una serie de servicios que forman parte de cualquier institución o empresa que no están relacionados con los clientes ni con la capa de negocio pero que son esenciales para su funcionamiento. Normalmente, en gestión empresarial, estos servicios son conocidos como los **servicios internos** de la empresa (*back office*) y son todos aquellos que dan soporte al negocio. Son, entre otros, los servicios informáticos, la gestión de recursos humanos, la formación de trabajadores, la contabilidad o las finanzas. El término *servicios internos* surge en contraposición con el de *servicios de cara al público* (*front office*), que es la parte de la empresa o de la institución que tiene contacto con los clientes o usuarios –los departamentos de ventas, distribución y mercadotecnia. En los servicios internos es donde se diseña, fabrica y gestiona la actividad objeto del negocio. Normalmente las tareas internas pueden ser realizadas directamente por la institución/empresa o pueden ser subcontratadas a proveedores externos (*outsourcing*), lo cual es muy habitual en los servicios de tecnologías de la información (servicios IT) –por ejemplo, en las empresas e instituciones es muy común la subcontratación de los servicios de correo electrónico y almacenamiento en la nube (*cloud*).

Existe una gran cantidad de argumentos en relación a los costes, ventajas tecnológicas sobre la competencia, reducción de los tiempos de inactividad o aumento de la productividad y hay gran cantidad de empresas que se dedican a este tipo de negocios a unos costes muy competitivos ofreciendo todo tipo de servicios IT: *service/help desk*, integración/consolidación de sistemas, integración de redes/comunicaciones, servicios de continuidad de IT/negocio, sistemas 24 × 7, impresoras y periféricos, correo electrónico y servicios de colaboración, plataformas de formación y comunicación empresarial, seguridad lógica y perimetral, monitorización y control contra ataques, gestión de activos, inteligencia de negocio, copias de seguridad, almacenamiento... También se debe considerar que hay una gran cantidad de detractores de la externalización de servicios IT debido a los problemas de dependencia que generan, la calidad del servicio provisto y los acuerdos de nivel de servicio –*Service Level Agreement* (SLA)–, que hay que tener en cuenta. Además, el proveedor no es un socio estratégico y, por tanto, no tiene los mismos intereses que la empresa o institución a la que sirve, por ello hay grandes dificultades en el seguimiento de los incrementos de costes: por servicios adicionales, por no adaptabilidad tecnológica, por recursos humanos deficientes –no profesionales por la reducción (temeraria) de costes–, por bloqueo de datos (prácticas *Lock-in*) en caso de desacuerdos/litigios o fin de la actividad (no esperada) de la empresa proveedora o por la pérdida de flexibilidad estratégica. Y así, una larga lista más [VROut].

Nuestro objetivo no es posicionarnos sobre las ventajas o inconvenientes de la externalización, sino mostrar una serie de servicios IT útiles y que cada vez más son más necesarios en el modelo de institución/empresa moderna y flexible, que es el prototipo de la empresa/institución competitiva actual. Estos servicios son parte de la estrategia de la empresa –cuyo seguimiento es una tarea que debe realizar el administrador de sistemas (*sysadmin*) y su equipo (por lo menos en las etapas de preproducción)– y forman parte de los servicios internos de la empresa. Un flujo de trabajo habitual de estos servicios es que se puedan montar como prueba de concepto bajo el departamento de IT de la empresa para luego definir con los responsables si debe ser un servicio interno y cuáles han de ser las características que debe cumplir para ponerlo en producción o si este se debe externalizar y debe ser llevado por un proveedor de servicios IT externo. En este último caso el trabajo del departamento de IT será determinar la mejor SLA para las necesidades del servicio y el coste que podrá ser asumido.

Para analizar los diferentes casos y hacer pruebas funcionales de algunas herramientas haremos la instalación desde un sistema virtualizado (Ubuntu 16.04 LTS) siguiendo las instrucciones del desarrollador. En otros casos, con el fin de reducir el tiempo de instalación, se ejecutarán imágenes en contenedores a través de Docker, ya sean de los propios desarrolladores del paquete o de la comunidad y que se encuentran disponibles en el *docker-hub*.

1. Sistema de gestión de contenidos

Un sistema de gestión de contenidos –*Content Management System (CMS)*– es un tipo de servidores que permite definir una estructura (*framework*) para la creación y gestión de contenidos como, por ejemplo, páginas web, blogs, wikis, publicaciones digitales, plataformas de enseñanza, difusión de contenidos, entre otros, con recursos multimedia y con delegación de los roles a diferentes usuarios, pero basado principalmente en la interacción y la interfaz centrada en el usuario. Normalmente están formados por una base de datos y una interfaz web desde la cual se pueden gestionar todos los recursos y administrar el servicio diferenciando el contenido del diseño y permitiendo que a los contenidos se les pueda aplicar una determinada plantilla sin tener que preocuparse por el formato. De esta manera un usuario sin grandes conocimientos de los recursos informáticos ni del servicio en cuestión podrá editar el contenido de forma fácil y difundirlo inmediatamente al resto de público. A continuación vamos a ver una serie gestores de contenidos representativos de aquellos que utilizan las empresas/instituciones con diferentes fines.

1.1. Drupal

Drupal es un gestor de contenidos *open source*, modular, multipropósito y configurable que permite publicar artículos, imágenes, archivos y que también ofrece la posibilidad de implantar una larga lista servicios añadidos como foros, encuestas, votaciones, blogs, etc. Su adaptabilidad y dinamismo le permiten fácilmente almacenar sus contenidos en archivos o en una base de datos en función de su tipo. Además, toda la interfaz de interacción (una vez configurada) se realiza mediante un entorno Web. Está escrito en PHP y funciona con un base de datos (generalmente MySQL). Su código es de calidad y se encuentra muy depurado, su diseño está pensado bajo criterios de usabilidad, cumple con los estándares web más representativos y dispone de una amplia comunidad que es muy activa en su desarrollo y mantenimiento. Esto explica que lo utilicen a nivel institucional/corporativo muchas entidades, empresas e instituciones representativas [Dru].

Entre los principales usos que se le puede dar destacan portales comunitarios, periódicos en línea, webs de comercio electrónico, foros de discusión, webs corporativas, aplicaciones de Intranet, sitios personales o blogs, aplicaciones de comercio electrónico, directorio de recursos, sitios de redes sociales, bibliotecas, etc. Para su instalación y configuración hemos utilizado una máquina virtual Ubuntu 16.04 y hemos dados los siguientes pasos:

¿Quién usa Drupal?

Muchas entidades, empresas e instituciones utilizan Drupal por sus ventajas: La Casa Blanca (USA), numerosos gobiernos (Reino Unido, Bélgica, Francia), universidades (Harvard, Stanford, Oxford), ONG (GreenPeace, Cruz Roja, Oxfam, Amnesty), artistas (Lady Gaga, Bruno Mars), empresas de comunicaciones (Warner Bros, Fox News, Weather Channel, NBC), entre otros.

- 1) `apt install php libapache2-mod-php php-mcrypt php-mysql`
- 2) `apt install php-mysql php-xml php-gd php-json php-curl`
- 3) `a2enmod rewrite`
- 4) `vi apache2.conf` y cambiamos *AllowOverride None* por *AllowOverride All* en la sección `<Directory /var/www/>... </Directory>`
- 5) Descargamos Drupal de <https://www.drupal.org/project/drupal/releases/8.1.7>
- 6) `mv drupal-8.1.7.tar.gz /var/www/html`
- 7) `cd /var/www/html/`
- 8) `tar xzvf drupal-8.1.7.tar.gz`
- 9) `mv drupal-8.1.7 drupal`
- 10) `cd drupal/sites/default`
- 11) `cp default.settings.php settings.php`
- 12) `cd /var/www/html`
- 13) `chown --recursive adminp:www-data drupal`
- 14) `chmod --recursive g+w drupal`
- 15) `service apache2 restart`
- 16) `apt install mysql-server`
- 17) `mysql -u root -p` introducimos el *passwd* dado durante la instalación y ejecutamos:

```
create database drupal;
create user 'drupal' identified by 'drupal';
grant all on drupal.* to 'drupal';
exit
```

18) Nos conectamos con un navegador a <http://localhost/drupal> e indicamos todos los datos (DB-drupal, *root*, *passwdDB_root*, correo electrónico, usuario administrador /*passwd* de Drupal, etc. Verificamos que no haya errores y si los hubiera los solucionamos antes de continuar la instalación.

19) Una vez se ha hecho la configuración ya se puede acceder y administrar el sitio cambiando su apariencia, contenidos, estructuras, bloques, menús, módulos, páginas, usuarios, roles y una larga lista de posibilidades. Finalmente se deben cambiar las protecciones de los archivos/directorios de configuración – sino dará una advertencia interna para subsanar el problema en el *Status Report* en la pestaña *Reports*– haciendo:

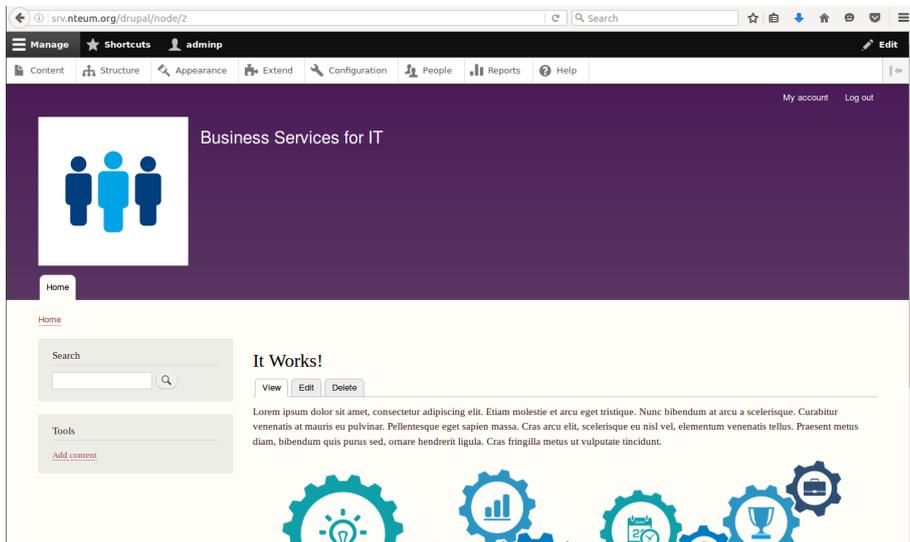
```
chmod 755 /var/www/html/drupal/sites/default
chmod 644 /var/www/html/drupal/sites/default/settings.php
```

Nota

Si se recarga la página *Status Report* de Drupal después de haber modificado los permisos veremos que la alerta ha desaparecido.

La figura 1 muestra una página inicial de nuestra instalación de Drupal con una página de prueba y algunos pequeños cambios en el logo, títulos, etc. para comprobar que toda la interfaz y gestión funcionan correctamente.

Figura 1



1.2. WordPress

WordPress es otro gestor de contenidos que si bien en un principio fue el gestor predominante para la creación de sitios de blog actualmente puede crear cualquier tipo de sitio y gestionar cualquier tipo de contenido. Sigue la terna de otros CMS (Apache+PHP+MySQL) bajo la categoría *open source* y está dentro de los CMS más utilizados y populares debido a su licencia, facilidad de uso y prestaciones (básicamente su extensibilidad a través de diferentes *plugins*). En la actualidad, junto con Drupal y Joomla!, es uno de los gestores de contenidos más utilizados. Por segmentos, mientras que Drupal está orientado hacia web complejas WordPress se usa más para la creación de web corporativas o de noticias (véase por ejemplo [MShare][CMSU]).

La haremos la instalación de WordPress a partir de un sistema Apache + PHP + MySQL (pasos 1, 2, 3 y 16 del subapartado 1.1), lo que se conoce como LAMP* o LEMP**:

```
1) mysql -u root -p
```

```
create database wordpress character set utf8 collate utf8_bin;
grant all privileges on wordpress.* to wordpressuser@localhost identified by "pssw";
flush privileges;
exit
```

reemplazando *pssw* por el *passwd* deseado.

```
2) cd /var/www/html
```

¿Quién usa WordPress?

Entre las instituciones y empresas más representativas que usan WordPress podemos destacar BBC America, The Rolling Stones, Blog Oficial de Yahoo!, SAP y NBC, entre otras.

* Linux, Apache, Mysql, PHP

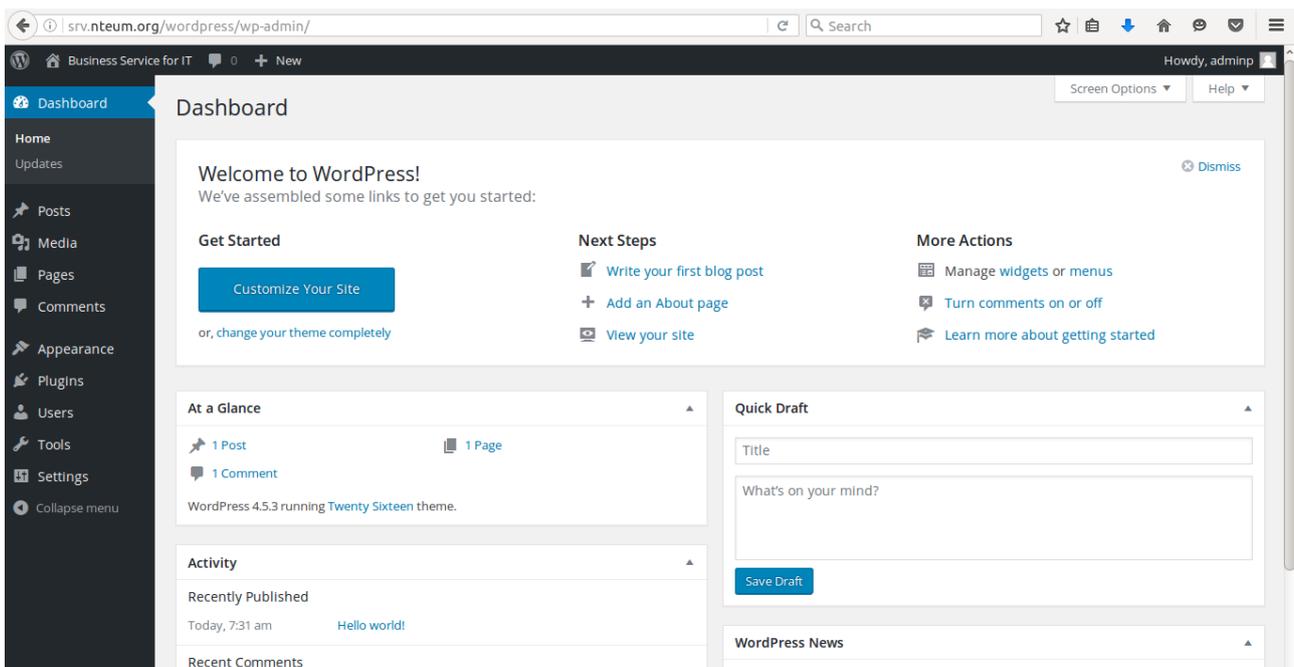
** Donde se cambia Apache por Nginx y la E es por la pronunciación de este último *engine-x*

- 3) `wget http://wordpress.org/latest.tar.gz`
- 4) `tar -xzvf latest.tar.gz`
- 5) `cp wordpress/wp-config-sample.php wordpress/wp-config.php`
- 6) Hacemos `vi wordpress/wp-config.php` y modificamos las líneas con los datos introducidos anteriormente:

```
define('DB_NAME', 'wordpress');  
define('DB_USER', 'wordpressuser');  
define('DB_PASSWORD', 'pssw');
```

- 7) En un navegador vamos a `http://site.dominio/wordpress` o la `ip-sitio/wordpress` y terminamos de complementar la información solicitada (título, usuario-admin/`passwd`, correo electrónico, privacidad). Ahora ya tendremos acceso al sitio después de autentificarnos y se nos mostrará un tablero (*dashboard*) similar al de la figura 2.

Figura 2

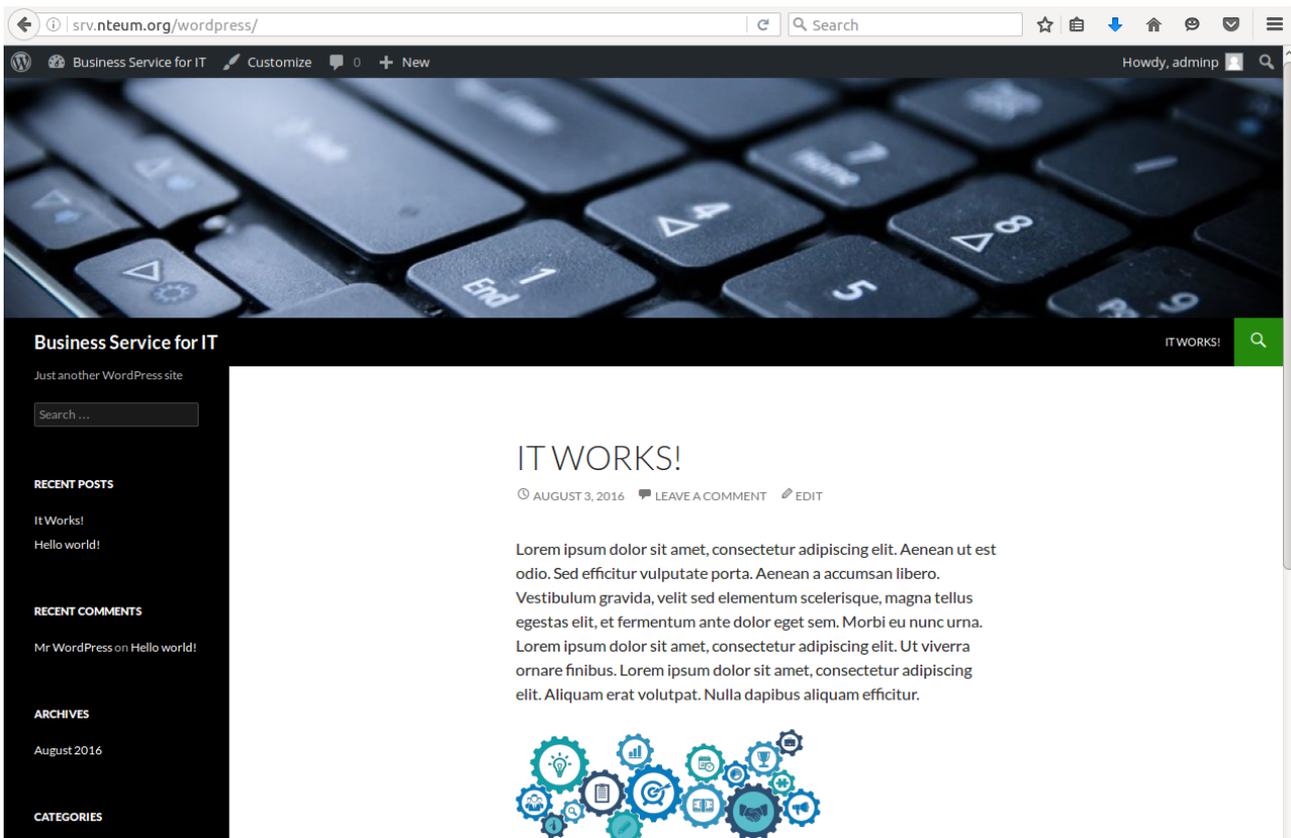


Veremos que si, por ejemplo, queremos subir imágenes nos dará un error de permisos que se soluciona con

```
chown -R root:www-data /var/www/html/wordpress  
chmod g+w /var/www/html/wordpress/wp-content
```

Después de algunas configuraciones el sitio podrá quedar por ejemplo tal como se muestra en la figura 3.

Figura 3



Existen otros gestores de contenidos con diferentes prestaciones/funcionalidades que son habitualmente utilizados como CMS, por ejemplo el ya mencionado Joomla! [Joo], Plone [Plo] (posicionado como *Enterprise CMS* y utilizado comúnmente para Intranets) o Liferay (versión Liferay Portal CE) [LR].

1.3. Moodle

Si bien Moodle no es exactamente un CMS, permite gestionar contenidos dentro de un entorno educativo virtual con el objetivo de facilitar el aprendizaje y la distribución de conocimientos en línea. Este tipo de plataformas se conoce como *Learning Content Management System* (LCMS) y su diseño está basado en las ideas del constructivismo y la enseñanza centrada en el estudiante. Frecuentemente se confunde LCMS con LMS (*Learning Management System*). Los LMS permiten administrar, distribuir y controlar las actividades de formación no presencial de una institución u organización (por ejemplo, gestionar usuarios, recursos, materiales, actividades, hacer el seguimiento del proceso de aprendizaje, evaluaciones, informes, etc.), pero no permiten administrar todo el proceso ligado a la autoría, cosa que sí incluyen los LCMS [Moo].

Este tipo de plataforma es muy utilizada en el ámbito de la formación en línea permitiendo a los formadores la gestión de cursos virtuales para sus alumnos

Liferay

Liferay, escrito en Java, es una *suite* orientada a CMS y basada en un servidor de aplicaciones –para la versión CE generalmente TomCat– y aprovecha toda la potencialidad de *servelets* y *portlets*.

(conocido como *e-learning*) o la utilización de un espacio virtual complementario a la formación presencial (conocido como *blended learning*). Comparte los requisitos de los CMS previamente analizados (LAMP/LEMP), dispone de una capa de abstracción de datos que permite flexibilidad en cuanto a la base de datos utilizada y posee una arquitectura extensible donde el diseño ha sido analizado con detalle especialmente en lo que a aspectos de seguridad se refiere.

Su instalación a partir de un LAMP/LEMP básico es [MInst] la siguiente:

- 1) `apt-get install git`
- 2) `cd /var/www/html`
- 3) `git clone git://git.moodle.org/moodle.git`
- 4) `cd moodle/`
- 5) `git branch -a`
- 6) `git branch --track MOODLE_31_STABLE origin/MOODLE_31_STABLE`
- 7) `git checkout MOODLE_31_STABLE` para quedarnos con la versión 3.1
- 8) Hacemos `vi /etc/mysql/mysql.conf.d/mysqld.cnf` y agregamos al final:

```
default_storage_engine = innodb
innodb_file_per_table = 1
innodb_file_format = Barracuda
```
- 9) `service mysql restart`
- 10) Con `mysql -u root -p` creamos la base de datos y parámetros.

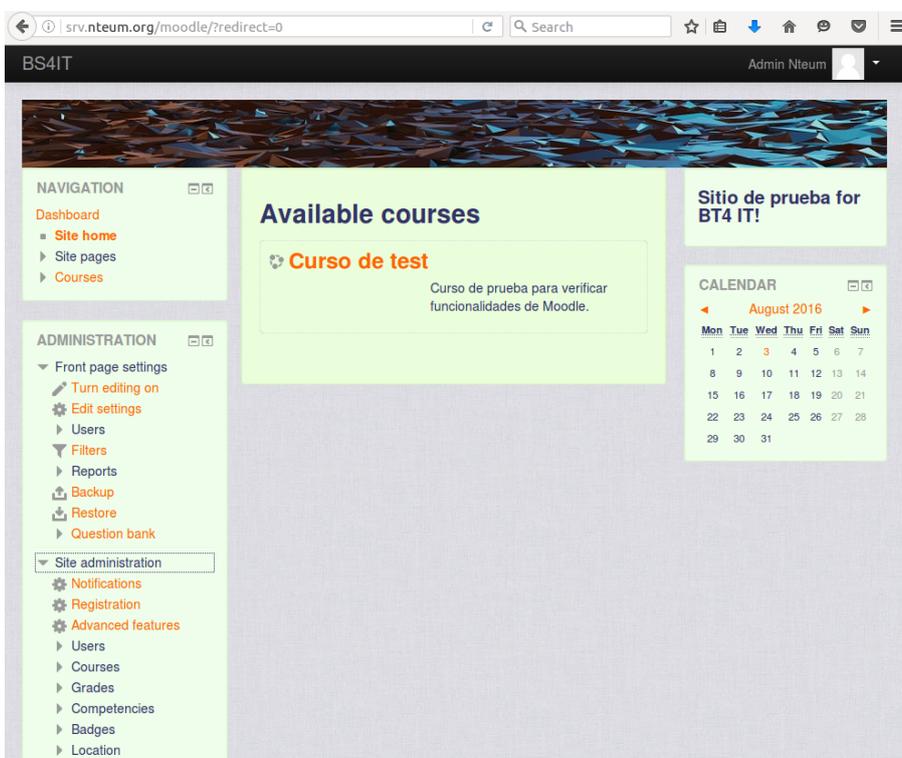
```
create database moodle default character set utf8 collate utf8_unicode_ci;
create user 'moodleuser'@'localhost' IDENTIFIED BY 'passw';
grant select , insert , update , delete , create , create temporary tables , drop , index , alter
on moodle.* to moodleuser@localhost IDENTIFIED BY 'passw';
quit;
```
- 11) `mkdir /var/www/moodledata`
- 12) `chown -R www-data /var/www/moodledata/`
- 13) `chmod -R 777 /var/www/moodledata`
- 14) `chown -R root:www-data moodle`
- 15) `chmod -R 777 /var/www/html/moodle`
- 16) `apt-get install php-zip php-xmllrpc php-soap php-intl php-mbstring`
- 17) `service restart apache2`

18) En un navegador vamos a *http://site.dominio/moodle* o la *ip-sitio/Moodle* y terminamos de complementar la información solicitada (*plugins*, extensiones faltante de php, título, usuario-admin/*passwd*, correo electrónico, ...). Se debe verificar que todas las extensiones necesarias de PHP estén instaladas y no hay que olvidarse de reiniciar Apache cada vez que se instala una. Una vez finalizada la instalación se podrá acceder al entorno para administrarlo, configurarlo y crear los cursos, usuarios/roles, herramientas... Finalmente, cuando se acabe la configuración, se deben cambiar las protecciones con

```
chmod -R 0755 /var/www/html/moodle
```

La figura 4 muestra la prueba de concepto con un curso configurado.

Figura 4



2. Monitorización

Si bien en el módulo de sintonización, monitorización y alta disponibilidad ya hemos visto una serie de herramientas para realizar la monitorización –como por ejemplo, Ganglia, Nagios/Icinga, Cactis (entre otros)– hay otras que son interesantes/complementarias dentro de un entorno institucional/empresarial para controlar las diferentes partes/áreas de los sistemas y la red. En este apartado veremos un conjunto de ellas. El administrador de sistemas deberá decidir cuáles se adaptan mejor a sus necesidades.

2.1. Graphite

Graphite es una herramienta *open source* (desde 2008) que permite monitorizar y graficar, con gran cantidad de detalles y funcionalidades, datos de diferentes monitores recogiéndolos, almacenándolos y visualizándolos como series temporales en tiempo real. La herramienta está compuesta por tres grandes componentes [Gr]:

- 1) **carbon**: el *daemon* que recoge *time-series data*.
- 2) **whisper**: una librería que permite procesar series temporales de datos (equivalente a RRD).
- 3) **graphite webapp**: una aplicación web en Django que dibuja los gráficos bajo demanda utilizando el entorno Cairo.

Graphite se integra con gran cantidad de *software* de monitorización como por ejemplo *collectd*, *Nagios*, *Sensu*, *Ganglia*, *Logstash*, *Sflow*, entre otros, y también permite reenviar métricas o visualizar datos dentro de otros entornos.

La forma más simple de probar la herramienta, como prueba de concepto, es utilizar **Synthesize** [Syn], que es un *script* de instalación y configuración totalmente automatizado para Graphite. Inicialmente fue desarrollado como una guía para nuevos usuarios que muestra cómo instalar manualmente la herramienta, pero se convirtió en un mecanismo de despliegue muy útil y permite hacerlo sobre un Ubuntu 14.04 LTS o se incluye un *Vagrantfile* para facilitar el despliegue de una maquina virtual para probarlo sin problemas. Synthesize incluye los siguientes componentes de *software* durante su instalación:

¿Quién usa Graphite?

Graphite se utiliza como herramienta de monitorización en grandes compañías, por ejemplo, GitHub, Salesforce, Reddit, Booking, Lyft, Simple o Etsy, entre otras.

- Graphite-web (Apache),
- Carbon,
- Whisper,
- Memcached (servicio de *caching* para acelerar las búsquedas),
- Statsite (implementación de altas prestaciones de StatsD)
- y Collectd (agente de métricas).

Para la opción de Vagrant, este ejecuta un pequeño Ubuntu 14.04 LTS Virtual-Box y una vez iniciado se ejecuta Synthesize (que dependiendo de la conexión tardará más o menos)*. En nuestro caso hemos preferido seguir la instalación en Ubuntu:

*Los pasos se pueden encontrar en [Syn].

1) `wget https://github.com/obfuscurity/synthesize/archive/master.zip`

2) `unzip master.zip`

3) `cd synthesize-master/`

4) Dado que el *script* no funciona si la versión no es igual a 14.04 y nosotros disponemos de una Ubuntu 16.04 como prueba funcional hemos editado el *script* `install` y hemos comentado la línea que controla esta acción (línea 18).

5) `./install` (solucionando los problemas/dependencias si las hay)

6) `reboot`

7) `cd /opt/graphite/webapp/graphite`

8) `python manage.py changepassword admin` para cambiar el `passwd` y acceder al `site`.

9) En un navegador vamos a `http://site.dominio/graphite` o a la `ip-sitio/graphite` y ya podremos acceder a crear gráficos de los monitores que incluye y diseñar un tablero (*dashboard*) con las gráficas de nuestro interés (si bien algunas opciones solo están permitidas cuando se está conectado con el usuario y *passwd* cambiados anteriormente).

10) De las pruebas realizadas no se han detectado mayores problemas en la utilización de una versión diferente de Ubuntu a la indicada. No obstante, para un entorno de producción se aconseja seguir la instalación indicada por el desarrollador [Gr]. La figura 5 muestra el tablero con un conjunto de gráficas de pruebas.

Otras herramientas interesantes en entornos complejos (además de las ya mencionadas) y que presentan otro tipo de funcionalidades, estrategias o complementos son Sensu [Sen], Shinken [Shi] y Splunk [Spl].

Figura 5



2.2. OpenNMS

OpenNMS es una plataforma *open source* de *IT monitoring* de nivel empresarial integrada que permite definir una solución de monitorización de red. Entre sus objetivos se incluye reducir el tiempo entre fallos detectando eventos que puedan afectar a la continuidad de los sistemas y la utilización de protocolos estándar de la industria de gestión de red, agentes y un sistema de aprovisionamiento programable.

Existe una gran comunidad OpenNMS que mantiene la plataforma y desarrolla conectores y nuevas interfaces para generar la integración de las soluciones de monitoreo de redes interoperables. Esta plataforma está basada en una arquitectura orientada a eventos que permite la integración de flujos de trabajo flexibles normalizando los mensajes específicos tanto de dispositivo como del proveedor y medidas de rendimiento específicos del protocolo [ONMS].

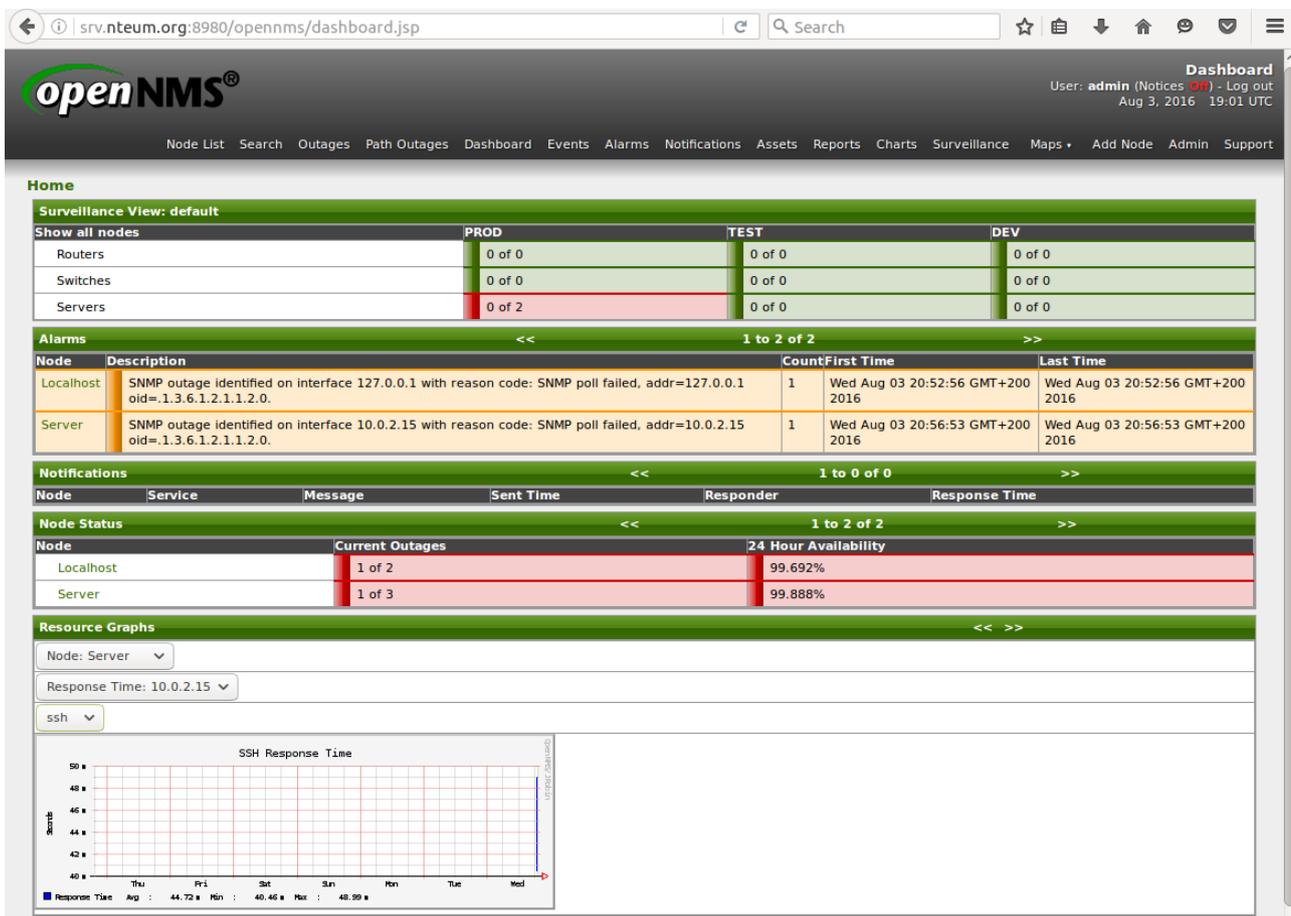
Para reducir el tiempo de instalación, que se puede seguir en [ONMSI], utilizaremos un entorno **Docker** para ver su funcionalidad sobre una Ubuntu 16.04 LTS. Utilizando el Docker de `study/opennms*` haremos:

*<https://hub.docker.com/r/study/opennms/>

- 1) `apt-get install docker.io`
- 2) `docker run --name my-postgres -e LC_ALL=C.UTF-8 -d postgres`

- 3) `docker run --name my-opennms --link my-postgres:postgres -p 8980:8980 -d 'study/opennms'`
- 4) Se puede verificar el funcionamiento con `docker ps` pararlo con `docker stop id_contenedor` y `docker rmi -f id_image` para eliminar el contenedor.
- 5) En un navegador vamos a `http://site.dominio:8980` o a `ip-sitio:8980` y con el usuario `|passwd admin` ya podremos acceder a la aplicación. Lo primero será crear nodos para monitorizar y ver el comportamiento a través del tablero. Es interesante analizar con detalle las múltiples opciones y cómo visualiza los dispositivos, alarmas y eventos. La figura 6 muestra una imagen del *dashboard* funcionando sobre Docker y con dos servidores monitorizados y el tiempo de respuesta del servicio `ssh`.

Figura 6



2.3. Zenoss (Zenoss Core)

Zenoss es una herramienta *open source* que se enmarca dentro de la categoría de *IT monitoring*. Ha sido diseñada para la gestión de redes y de servidores y está basada en el servidor de aplicaciones Zope con una interfaz web que permite a los administradores de sistemas monitorizar la disponibilidad, el

inventario y la configuración, así como aspectos funcionales y eventos. La versión Core (hasta 1.000 dispositivos) permite la utilización de SNMP para dispositivos que soporten este protocolo, monitorización de servicios de red (HTTP, POP3, NNTP, SNMP, FTP), *host* (cpu, utilización de disco) en la mayoría de los sistemas operativos de red, rendimiento a través de serie temporales de datos, alertas, detección automática de recursos en red y cambios en su configuración, notificaciones basadas en reglas y calendarios y además soporta el formato de los *plugins* de Nagios. [Zen]

Para probar esta aplicación utilizaremos la imagen de Docker *dfritz/zenoss** que bajaremos con `docker pull dfritz/zenoss` y cuando finalice la descarga pondremos en marcha con

[*https://hub.docker.com/r/dfritz/zenoss/](https://hub.docker.com/r/dfritz/zenoss/)

```
docker run --name my-zenoss -p 80:80 -d 'dfritz/zenoss'
```

Finalmente haremos un `docker ps` y con el *container_id* haremos un `docker attach container_id` que iniciará el proceso de poner en marcha todos los *daemons*. Por último, podremos conectarnos a la IP del host (ya que hemos mapeado el puerto 80 de Docker en el *host*). Se debe tener en cuenta que es un paquete complejo y que la maquina virtual necesitará por lo menos 2,5 Gb de memoria para que pueda funcionar aceptablemente. La figura 7 muestra una de las vistas que permite la aplicación.

Figura 7

The screenshot displays the Zenoss CORE web interface. The top navigation bar includes 'Dashboard', 'Events', 'Infrastructure', 'Reports', and 'Advanced'. The main content area shows the configuration for a device with ID '10.0.2.15'. The device status is 'Up' and the production state is 'Production'. The configuration panels include:

- Overview:** Lists various monitoring templates and properties.
- Device Information:** Shows fields for Device ID (10.0.2.15), Uptime (Unknown), First Seen (2016/08/03 20:33:39), Last Change (2016/08/03 20:51:19), Model Time (Not Modeled), Locking (Unlocked), and Memory/Swap (Unknown/Unknown).
- Device Configuration:** Includes fields for Device Title (10.0.2.15), Production State (Production), Priority (Normal), Tag, and Serial Number.
- Rack Slot Configuration:** Includes fields for Rack Slot (0), Collector (localhost), Hardware Manufacturer (None), Hardware Model (None), OS Manufacturer (None), and OS Model (None).

Dentro los paquetes de monitorización y observación de redes para infraestructuras complejas también se pueden considerar Observium [Obs], Pandora [Pan] o RackTables [Rac] que si bien son potentes necesitan una infraestructura apropiada. A veces puede ser complicado instalarlo en una máquina virtual o en contenedores. No obstante se pueden hacer pruebas con los contenedores disponibles en DockerHub.

3. Gestión de proyectos: Redmine

La gestión de proyectos en una empresa o institución es una actividad esencial sobre todo cuando se debe hacer planificación de recursos multidisciplinares, control de costos y seguimiento de tareas para alcanzar unos objetivos indicados en un tiempo acordado. Hay dos partes importantes en la gestión de un proyecto: el primero es alcanzar los objetivos dentro de unas limitaciones/restricciones predefinidas como son el tiempo, la calidad y el presupuesto; y, el segundo, es optimizar la asignación de recursos para alcanzar los objetivos definidos.

Redmine es una herramienta *open source* (escrito en Ruby on Rails) para la gestión de proyectos que incluye, entre otras opciones, un sistema de seguimiento de incidentes, un calendario de actividades, diagramas de Gantt para la representación visual de la línea del tiempo de los proyectos, wiki, foro, repositorio de control de versiones, RSS, control de flujo de trabajo basado en roles e integración con correo electrónico[Red].

Para la prueba de concepto de Redmine es posible instalarla siguiendo las instrucciones de los desarrolladores [RedI] o utilizando el contenedor Docker oficial*. Para ello hacemos

```
docker run -d --name my-redmine2 -p 3000:3000 redmine
```

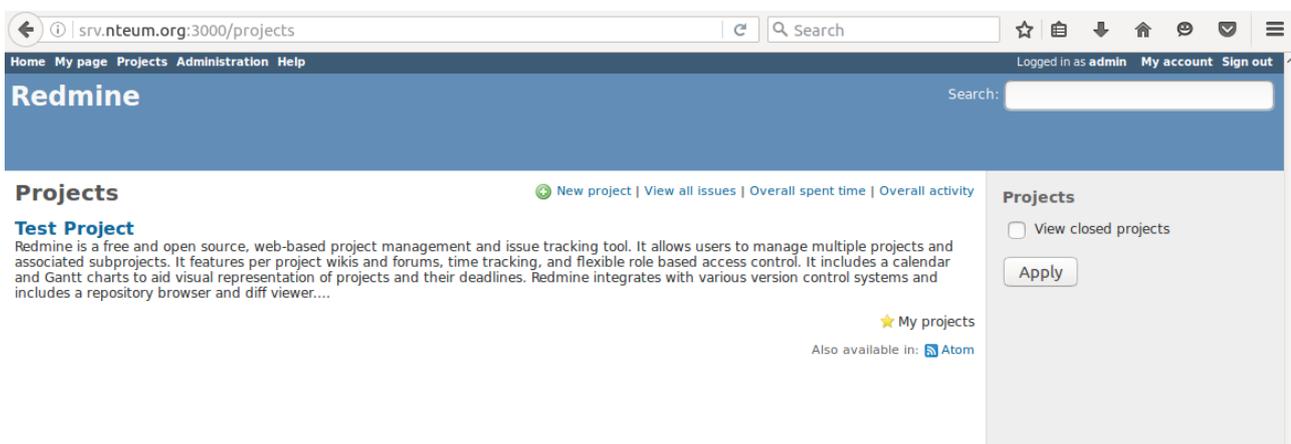
y luego abrimos un navegador en `http://site.dominio:3000` o la `ip-sitio:3000` y con el usuario `passwd admin` ya podremos acceder a la aplicación. La figura 8 muestra una imagen de esta ejecución con un proyecto de ejemplo.

*https://hub.docker.com/_/redmine/

Otras herramientas

Como otras herramientas similares a Redmine son Trac [Tra] y ProjectPier [PPi].

Figura 8



4. Cloud Storage

Además de las aplicaciones mostradas en otros módulos como OwnCloud, Pydio es una alternativa *open source* a Dropbox o box.net que permite el intercambio de archivos. Está centrada en el usuario y orientada a empresas y organizaciones, ya que permite controlar los datos, que se encuentran en el servidor corporativo, y su privacidad. Dispone de cliente para los dispositivos móviles más comunes y para algunas aplicaciones como por ejemplo un *plugin* para Thunderbird o para los principales CMS, y permite una funcionalidad similar a los proyectos comerciales mencionados anteriormente. Su instalación es sumamente simple [PyI] y se puede hacer de diferentes formas, pero en nuestro caso para hacer un test funcional utilizaremos el contenedor `kdelfour/pydio-docker*`. Para ello ejecutamos

```
docker run -it -d -p 80:80 -p 443:443 kdelfour/pydio-docker
```

y si se desea agregar un directorio compartido con el *host* haremos `-v /your-path/files:/var/www/pydio-core/data/files/ -v /your-path/personal:/var/www/pydio-core/data/personal/ -v /your-path/mysql:/var/lib/mysql/`. Por ejemplo (hay que tener en cuenta que el comando es una única línea y que se debe reemplazar *your-alguna-cosa* por los valores adecuados):

```
docker run -it -d -p 80:80 -p 443:443
-v /your-path/files:/var/www/pydio-core/data/files/
-v /your-path/personal:/var/www/pydio-core/data/personal/
-v /your-path/mysql:/var/lib/mysql/
kdelfour/pydio-docker
```

El servidor `mysql` está incluido con estos parámetros (que serán solicitados al inicio de la configuración de Pydio):

```
url : localhost
database name : pydio
user name : pydio
user password : pydio
```

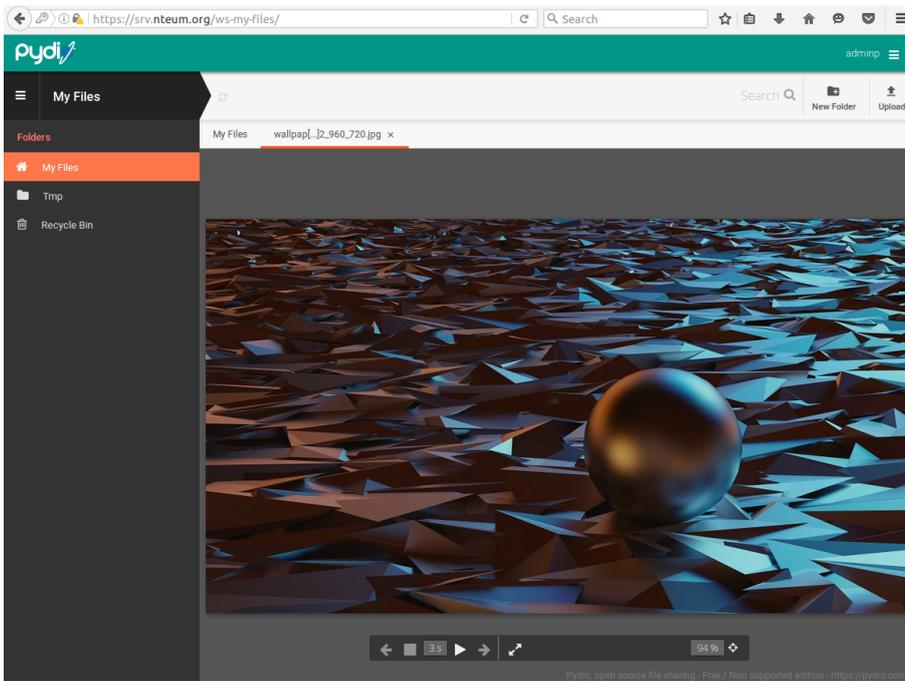
Finalmente abrimos un navegador en `https://site.dominio` o en `https://ip-sitio` y aceptamos el certificado, iniciamos la configuración (base de datos, usuario, etc.) y ya podremos acceder a la aplicación. La figura 9 muestra una imagen de esta ejecución con unos archivos de ejemplo.

Pydio

Pydio son las iniciales de *Put your data in Orbit*.

*<https://hub.docker.com/r/kdelfour/pydio-docker/>

Figura 9



SynsAny

Un proyecto interesante dentro de la misma categoría es SynsAny que permite sincronizar y compartir en forma segura archivos en cualquier tipo de almacenamiento [Syn].

5. Sistemas operativos virtuales o Cloud OS

Con el auge del cómputo en la nube (*cloud computing*) se han renovado las fuerzas en los sistemas operativos *online* o virtualizados. El objetivo de estos sistemas operativos es poder realizar todas las tareas que haríamos en un ordenador local con un sistema operativo instalado, pero sobre un recurso remoto (*cloud*) y solamente disponiendo de un navegador, una conexión a Internet y sin la necesidad de que el recurso local sea muy potente, ya que la capacidad de procesamiento será la de recurso remoto. En este caso en el cliente (navegador) solo se visualizará el resultado. Dado que el sistema remoto puede ser escalable la potencialidad de estos sistemas es muy grande, ya que además el usuario final no debe preocuparse por la instalación y mantenimiento. De esta manera los gestores de IT no deben preocuparse por los dispositivos que sus clientes/empleados utilizan para el acceso al servicio remoto. Dentro de estos sistemas podemos mencionar:

- **Chromium OS.** Es un proyecto *open source* para construir un sistema operativo que pueda ofrecer una alternativa rápida, sencilla y segura para las personas que pasan la mayor parte de su tiempo navegando por la web. Se puede descargar y compilar a partir del código fuente y funciona sobre microprocesadores con tecnología x86, x64 o ARM. Este proyecto es la semilla del Chrome OS desarrollado por Google (desde 2009 y bajo versión propietaria) instalada sobre el hardware específico de fabricantes asociados (Samsung, LG. . .) orientado a miniportátiles o también llamados Chromebooks [Chr].
- **EyeOS.** Inicialmente fue un proyecto *open source* pero desde 2012 tiene licencia comercial. A través de una interfaz web se proporciona un escritorio completo desde la nube con gestión de archivos, herramientas de gestión de la información personal, herramientas colaborativas y aplicaciones de la compañía. Combina la potencia de HTML5, PHP, AJAX y JavaScript para crear un entorno gráfico de escritorio (solo se requiere un navegador que soporte AJAX, Java y Adobe Flash). Es interesante acceder al nuevo proyecto (*fork* del anterior) mantenido por la comunidad denominado OneEye [OE].
- **OS.js.** Es un sistema operativo web *open source* escrito en JavaScript. Dentro del navegador el usuario tiene todo un entorno como si de un sistema operativo real se tratara (gestor de ventanas, API de aplicaciones, GUI, configuraciones, aplicaciones. . .). Como está desarrollado en Javascript, funciona sobre cualquier navegador actual y puede ser instalado sobre todas las

WebOS

A veces a los sistemas operativos virtuales se les llama también WebOS, pero se suele evitar el término ya que identifica a un proyecto concreto.

plataformas. El entorno está inspirado en los sistemas Linux pero diseñado para que sea fácil y rápido acceder a las aplicaciones/configuraciones. Con el sistema de archivos virtual el usuario puede subir y bajar archivos o modificarlos/moverlos a través del *cloud* como por ejemplo Google Drive, Dropbox o OneDrive o el propio OwnCloud/Pydio corporativo. Entre las aplicaciones habituales que soporta están: gestor de archivos, reproductor de música y multimedia, editor de imágenes, calculadora, editor de texto, visor de PDF, chat XMPP, Google Mail, Google Contacts, Tetris and Wolfenstein3D. Dispone además de una API en Javascript muy flexible que permite modificar las funcionalidades actuales o crear nuevas, así como también nuevas aplicaciones [OS.js].

- 1) Su instalación se puede hacer desde diferentes medios/entornos, pero en este caso utilizaremos una instalación sobre Ubuntu 16.04. Es sumamente fácil. Solo hay que ejecutar:

```
cd /root; curl -sS https://os.js.org/installer | sh
```

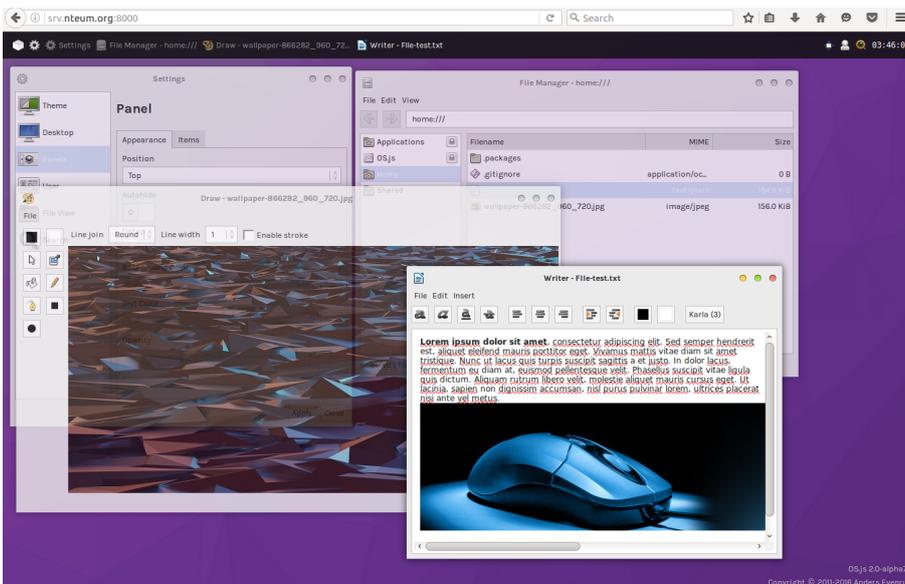
- 2) Después de unos minutos podremos ponerlo en marcha ejecutando

```
OS.js/bin/start-dist.sh
```

- 3) Luego podremos conectarnos a <http://srv.nteum.org:8000> y ya podremos acceder al sistema remoto.

La figura 10 muestra una ejecución con cuatro ventanas abiertas: preferencias, gestor de archivos, editor de imágenes/dibujo y editor de texto.

Figura 10



Otros ejemplos

Como otros ejemplos representativos de *Cloud OS* podemos mencionar Cloud OS [COS], ONOS [Ono], Lucid [Luc].

6. Sitios de colaboración

Este tipo de herramientas son la base para la colaboración entre personas de un grupo/departamento/empresa que tienen intereses comunes y comparten recursos y objetivos dentro del mismo entorno. Generalmente se basan en un entorno web que dispone de un conjunto de herramientas tales como correo electrónico, calendario, agenda, notas, tareas, espacio para compartir archivos, chat... Este tipo de herramienta es conocido también como Groupware por la definición de C. A. Ellis (1980) que definió con este término

“computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment”.

Existen un conjunto de herramientas que se pueden evaluar para buscar la que se adapte mejor a las necesidades del grupo. Entre ellas se encuentran Citadel [Cit], Zimbra [Zim], eGroupware [eGr].

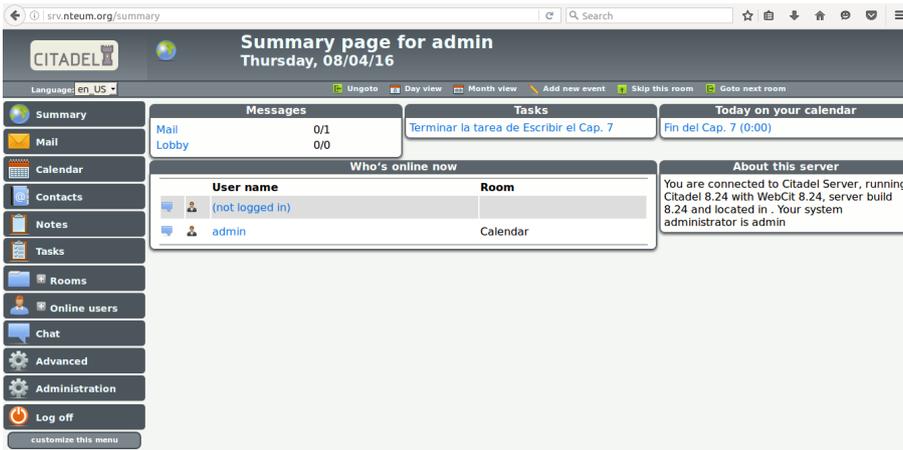
Citadel (a veces llamado Citadel/UX) es una *suite* de colaboración (mensajería y trabajo en grupo) que descende de la familia Citadel (popular en las décadas de los 80 y 90) como una plataforma de sistema de tablón de anuncios (BBS). Es una herramienta *open source* y soporta una serie de herramientas de comunicación, colaboración, protocolos de Internet (SMTP, IMAP, Tamiz, POP3, GroupDAV y XMPP) y OpenSSL para proteger la privacidad. Como prueba de concepto utilizaremos un contenedor Docker del DockerHub `million12/citadel*` ejecutando (hay que tener en cuenta que el comando es en una única línea y que se deben reemplazar los valores de `--env` por los apropiados):

[*https://hub.docker.com/r/million12/citadel/](https://hub.docker.com/r/million12/citadel/)

```
docker run -d -h mail.nteum.org --name citadel --dns 8.8.8.8 -p 25:25
-p 110:110 -p 143:143 -p 465:465 -p 587:587 -p 993:993 -p 995:995 -p 80:8080
-p 10022:22 --env="ROOT_PASS=psswd-root" --env="PASSWORD=passw"
--env="DOMAIN=nteum.org" --env="ATOM_SUPPORT=true" million12/citadel
```

Donde `ROOT_PASS = root user password` (si no se especifica la imagen generará uno aleatorio que puede ser obtenido con `docker logs docker_name`), `PASSWORD = admin@nteum.org passwd` para la interfaz web del usuario `admin`, `DOMAIN = dominio especificado por el usuario`, `ATOM_SUPPORT=True` soporte para un editor Atom. Para acceder hacemos simplemente `http://srv.nteum.org` o la ip del `host` con el usuario `admin` y el `passwd` indicado. En la figura 11 se puede ver una imagen de la herramienta funcionando en el contenedor y con diferentes eventos mostrando alguna de su funcionalidad (mail, tareas, calendario, contactos).

Figura 11



La figura 12 muestra el funcionamiento de eGroupware (en la creación de un *site*) a través del contenedor `systemsector/egroupware*` ejecutando

[*https://hub.docker.com/r/systemsector/egroupware/](https://hub.docker.com/r/systemsector/egroupware/)

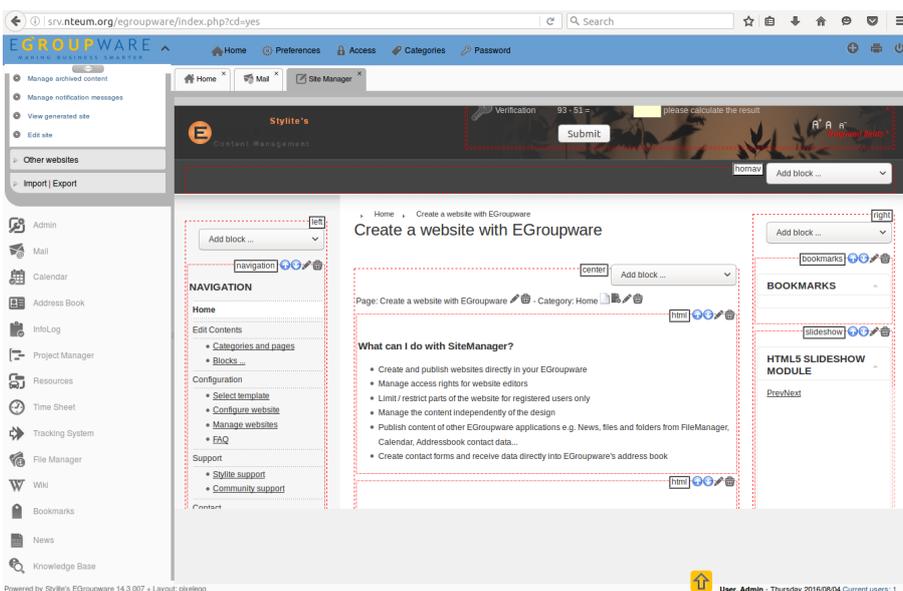
```
docker pull systemsector/egroupware:latest
```

Y a continuación

```
docker run -i -t -p 80:80 systemsector/egroupware:latest /bin/bash
```

y accediendo `http://srv.nteum.org/egroupware` con el usuario `sysop` y el `passwd password`. Como se puede observar el conjunto de herramientas es similar, aunque este dispone de algunas más.

Figura 12



7. ERP

Los sistemas de planificación de recursos empresariales –*Enterprise Resource Planning* (ERP)– son sistemas de información gerenciales que integran y manejan diversos aspectos de las operaciones de producción y distribución de una empresa cuando produce bienes o servicios. Básicamente es un *software* complejo que maneja la producción, logística, distribución, inventario, envíos, facturas y contabilidad de forma modular. Algunos ERP también soportan extensiones para ventas, entregas, pagos, producción, calidad de administración y la administración de recursos humanos. Los sistemas ERP habitualmente forman parte de los servicios internos (*back office*) de las empresas, ya que el cliente/público general no tienen acceso a él, pero sí cumple con el rol de *Supplier Relationship Management* (SRM), ya que trata con los proveedores (más allá de la relación administrativa). No obstante, los ERP actuales tienden a ocupar todos los espacios de una organización incluyendo actividades propias de un *Customer Relationship Management* (CRM).

Odoo es un sistema de ERP integrado de código abierto de Odoo S.A y sus desarrolladores afirman que es equivalente a otros *software* propietarios (por ejemplo, SAP ERP o Microsoft Dynamics). Esta aplicación, escrita en Python y que utiliza PostgreSQL como gestor de bases de datos, tiene una arquitectura web con múltiples aplicaciones cliente (módulos) que se comunican con el servidor a través de interfaces XML-RPC. Los módulos son carpetas con código en Python más archivos XML que definen la estructura de los datos, formularios, informes, menús, procedimientos, flujos de trabajo, etc., y que permiten ser adaptados las necesidades del cliente. En sus reportes se integra con OpenOffice.org, lo que permite personalizar los informes y entre sus principales módulos se puede enumerar la gestión de: compraventa, CRM, gestión de proyectos, almacenes, manufactura, contabilidad analítica y financiera, puntos de venta, activos, recursos humanos, inventario, marketing, flujos de trabajo entre otros [Odoo].

Las figuras 13 y 14 muestran una prueba funcional a través del contenedor oficial `odoo`* donde se muestra los módulos que se han de gestionar y una prueba del módulo inventario (con datos ficticios cargados durante la inicialización del programa).

Odoo era conocido anteriormente como OpenERP y antes TinyERP.

*https://hub.docker.com/_/odoo/

Figura 13

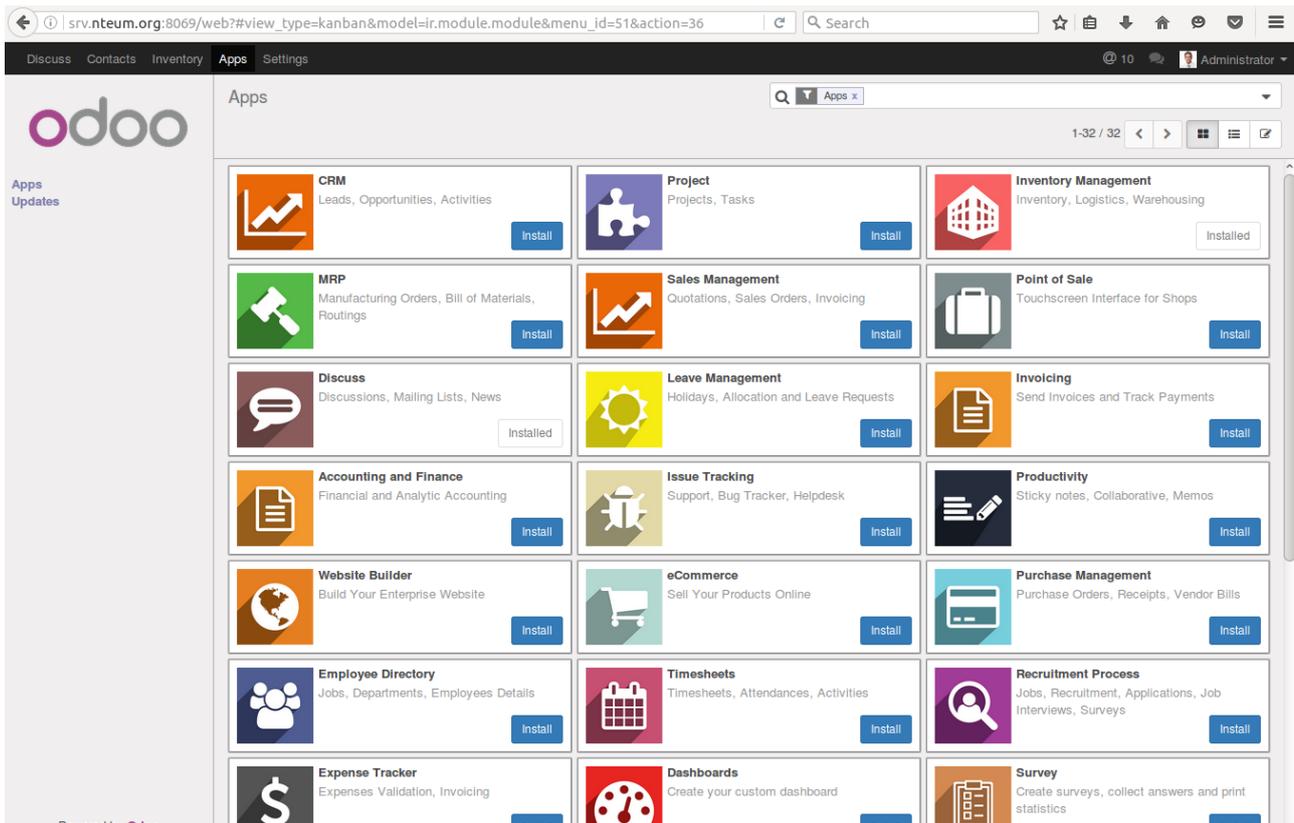
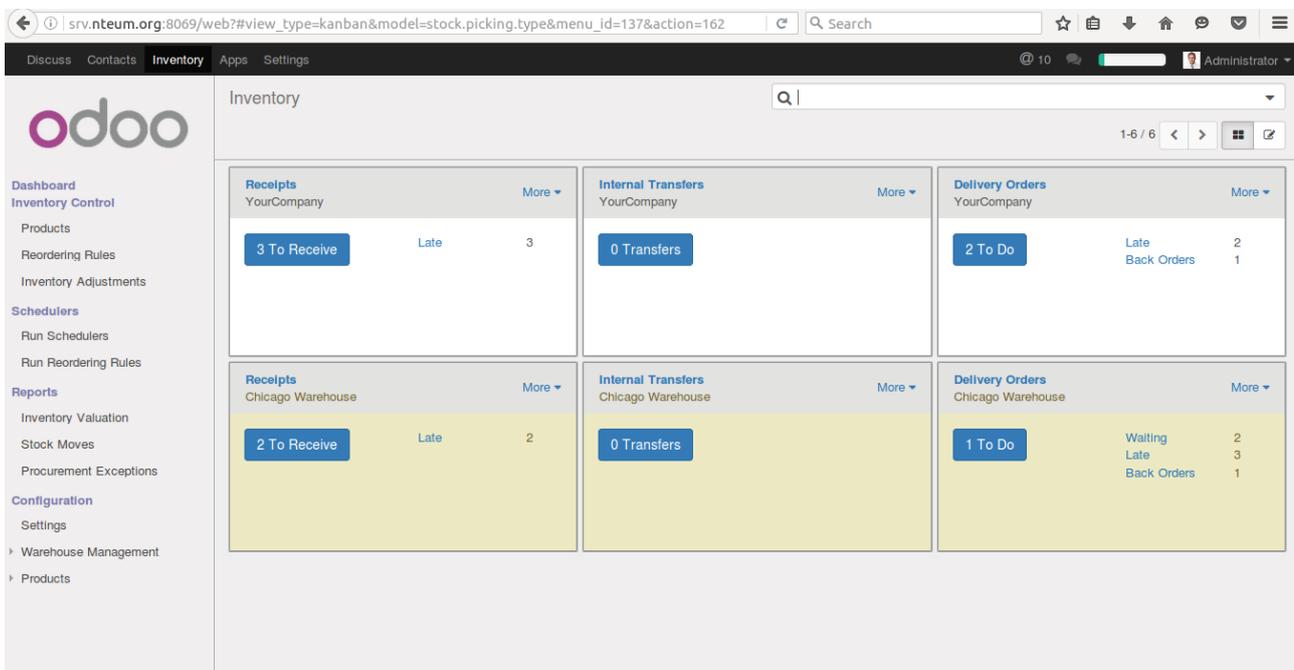


Figura 14



Su ejecución es primero cargando un contenedor de Postgres y luego el propio de Odoo enlazado con el anterior:

1) `docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo --name db postgres:9.4`

2) `docker run -p 8069:8069 --name odoo --link db:db -t odoo`

3) Luego de puestos en marcha se debe conectar un navegador al *host* y el puerto 8069 (en nuestro caso a <http://srv.nteum.org:8069>) o la ip del *host* y finalizar la configuración de la base de datos demos/mail-usuario/passwd e indicar si se desean cargar datos de demostración. El acceso luego será a través del mail/*passwd* indicado.

8. Otras aplicaciones útiles en el *back office*

Como se ha podido comprobar existen un gran conjunto de aplicaciones de gran calidad y *open source* que permiten montar diferentes estrategias de *back office* y dar servicio a los usuarios de una empresa o institución. En los apartados anteriores se han mostrado algunas de ellas, pero la lista no termina aquí y en este apartado se mencionarán algunas más para que el administrador de sistemas las tenga en cuenta en la definición de los servicios que se deben prestar a los usuarios. La dinámica de instalación y pruebas siempre es la misma y se debe iniciar en una máquina virtual o a través de los contenedores Docker de los propios desarrolladores o los de la comunidad, que permiten evaluar una herramienta muy rápidamente sin todos los inconvenientes de dependencias, bases de datos, librerías, versiones, etc., que algunas de ellas tienen. Siempre que sea posible se recomienda utilizar Docker por su flexibilidad y rapidez en la puesta en marcha de las nuevas herramientas. Hay que recordar siempre que son pruebas de concepto y en ningún caso instalaciones de producción.

Entre una gran lista de herramientas posibles podemos mencionar (en orden alfabético):

- **Alfresco** (categoría *open platform for business software*). Es una plataforma de gestión de contenidos *open source* desarrollado en Java, con estándares abiertos y de escala empresarial. Dispone de diferentes variantes y modalidades de servicio/acceso y una de ellas es Alfresco Community Edition, que es abierta a la comunidad. Su diseño es adecuado para entornos que requieren un alto grado de modularidad y rendimiento escalable e incluye un repositorio de contenidos, un *framework* de portal web para administrar y usar contenido estándar en portales, una interfaz CIFS (SMB) para compatibilidad de sistemas de archivos en Windows, un sistema de administración de contenido web, capacidad de virtualizar aplicaciones web y sitios estáticos vía Apache Tomcat, búsquedas vía el motor Apache Solr-Lucene y flujo de trabajo en jBPM. Es una plataforma con muchas funcionalidades, pero compleja y su curva de aprendizaje, en general, tiene una baja pendiente [Alf].
- **BonitaBPM** (categoría *business process management & workflow*). Es una plataforma *open source* para la gestión de procesos de negocio (BPM) y realización de *workflows* desarrollada inicialmente por NIRCRC (Francia) y actualmente Bonitasof.com. Su arquitectura se compone de tres partes:
 - 1) **Bonita Studio**, que permite modificar gráficamente los procesos de negocio siguiendo el estándar BPMN y también conectar procesos a otros

sistemas de información (mensajería, ERP, ECM, bases de datos...) para generar una aplicación de negocios autónoma;

2) **Bonita BPM Engine**, que es una API JAVA para interactuar con los procesos (utiliza Hibernate), y

3) **Bonita User Experience**, que es un portal web para gestionar (en una interfaz *webmail-like*) las tareas y procesos bajo control (basado en AngularJS) [BBPM].

- **Elasticsearch** (categoría *search and analytics engine*). Es un servidor de búsqueda basado en Lucene que provee un motor de búsqueda de texto completo, distribuido y con capacidad de multitenencia con una interfaz web RESTful y con documentos JSON. Elasticsearch está desarrollado en Java y está publicado como *open source* bajo licencia Apache [Ela].
- **Elastix** (categoría *open source communications*). Es una distribución libre de servidor de comunicaciones unificadas que integra en un solo paquete voIP PBX (*Private Branch Exchange* o central telefónica), fax, mensajería instantánea, correo electrónico y herramientas de colaboración y está basado en cuatro grandes programas como Asterisk, Hylafax, Openfire y Postfix que proveen las funcionalidades de PBX, fax, mensajería instantánea y correo electrónico respectivamente. La plataforma se ejecuta sobre CentOS e incorpora un módulo de *Call Center*, que interactúa con una consola de agente a través de un protocolo abierto denominado *Elastix Call Center Protocol* (ECCP) que permite la comunicación con consolas desarrolladas por terceros diseñadas para actuar como agente o supervisor [Etix].
- **Elgg** (categoría *Social Network Engine*). Es una plataforma *open source* de servicios de red social que ofrece blog, trabajo en red, comunidad, recolección de noticias vía *feeds* e intercambio de archivos. Todo puede ser compartido entre los usuarios, utilizando los controles de acceso y puede ser catalogado mediante etiquetas (*tags*) [Elg].
- **LifeRay Portal CE** (categoría *free and open source enterprise portal*). Es una plataforma *open source* para proveer las funcionalidades de portal empresarial y se utiliza principalmente para potenciar las intranets y extranets corporativas. Esta plataforma está escrita en Java e incluye un sistema de gestión de contenidos web integrado que permite a los usuarios crear sitios web y portales como un conjunto de temas, páginas, *portlets/gadgets* y una navegación común. Algunos desarrolladores lo definen como un *framework* de gestión de contenido o de aplicación web con una arquitectura extensible mediante *plugins* y que incorpora soporte para *portlets* en PHP y Ruby. Esta aplicación funciona con diferentes servidores de aplicaciones, pero la versión CE está asociada con TomCat como *servlet container* [Lif].
- **OpenBravoERP** (categoría ERP). Es una aplicación de ERP basada en arquitectura web orientada a la PYME bajo una licencia Openbravo Public Li-

cense basada en Mozilla PL. Esta aplicación permite automatizar y registrar los procesos de negocio más comunes como ventas, compras, fabricación, proyectos, finanzas, MRP, entre otros [OBr].

- **OTRS** (categoría *Ticketing System*). Es una plataforma web *open source* para asignar identificadores únicos (llamados *tickets*) a solicitudes de servicio o de información, de forma que se facilita el seguimiento y el manejo de incidencias así como cualquier otra interacción con sus clientes o usuarios [OTRS].
- **Piwik** (categoría *Open Source Liberating Analytics*). Es una aplicación *open source* de análisis web sobre PHP-MySQL que permite analizar el seguimiento de las visitas en línea a uno o más sitios web generando informes detallados sobre estos. Es muy popular y cuenta con un grupo de desarrolladores muy activo que permite tener un mantenimiento periódico y nuevas prestaciones [Piw].
- **PrestaShop** (categoría *ecommerce CMS*). Es un sistema gestor de contenidos (CMS) *open source* orientado a la creación y gestión de tiendas en línea de comercio electrónico. Es muy popular y tiene gran aceptación. Es compatible con pasarelas de pago como DirecPay, Google CheckOut, Authorize.net, Moneybookers o PayPal, pero pueden integrarse otros módulos de pago adicionales disponibles en el mercado [PrS].
- **Proxmox VE** (categoría *virtualization*). Es una plataforma *open source* que ofrece un entorno de virtualización basado en Debian que permite el despliegue de *virtual machines* sobre KVM y *containers* sobre LXC e incluye una consola web de gestión, una CLI y una API REST para integrarse con herramientas de terceros. Puede ser instalado desde la ISO o bien desde diferentes distribuciones de Linux, por ejemplo, Debian Jessie [Pro].
- **SugarCRM CE** (categoría CRM). Es un sistema para la administración de la relación con los clientes (CRM) basado en LAMP y desarrollado por la empresa SugarCRM (California). Dispone de diferentes ediciones –una de ellas la versión *community* (SugarCRM CE)– y soporta la gestión de ventas, oportunidades, contactos de negocios entre otras funcionalidades [Sug].

Obviamente la lista no acaba aquí y solo es un ejemplo de la disponibilidad de herramientas y plataformas en la modalidad de *open source* para satisfacer las necesidades del *back office* o la estrategia de negocio de la empresa/institución.

Actividades

Dado que en este módulo se presenta un conjunto amplio de herramientas, se sugiere como actividad que el alumno, para cada categoría de herramientas, realice la instalación de cada tipo valorando la funcionalidad y características de cada plataforma desplegada y utilizando dos tipos de instalaciones:

- 1) desde el contenedor oficial (si existe) o de la comunidad y
- 2) sobre una máquina virtual cumpliendo los requerimientos de los desarrolladores.

Es importante que el alumno documente para un análisis posterior los procesos de instalación y extraiga conclusiones sobre el proceso y dificultades/soluciones encontradas.

Bibliografía

Todo los enlaces fueron visitados en agosto de 2016.

- [Alf] Alfresco CE. Plataforma de convergencia entre ECM y BMP (Enterprise Content Management-Business Process Management). <<https://www.alfresco.com/es/products/community/download>>
- [BBPM] BonitaBPM. <<http://www.bonitasoft.com/downloads-v2?version=7.3.0&os=linux> https://hub.docker.com/_/bonita/>
- [Chr] Chromium OS. <<https://www.chromium.org/chromium-os>>
- [Cit] Citadel. <<http://www.citadel.org/doku.php>>
- [CMSU] CMS Usage. <<https://cmsusage.com/>>
- [COS] ClouOS. <http://cloudos.internet.de/installation-manual_14-9>
- [Dru] Drupal. <<https://www.drupal.org/>>
- [eGr] eGroupware. <<http://www.egroupware.org/products/installation-on-your-own-server-free-version.html>>
- [Ela] ElasticSearch. <<https://www.elastic.co/downloads>>
<https://hub.docker.com/_/elasticsearch/>
- [Elg] Elgg. Powerful open source social networking engine. <<https://elgg.org/download.php>>
- [Etix] Elastix. Unified communications server. <<http://www.elastix.com/en/downloads/>>
- [Gr] Graphite Overview. <<http://graphite.readthedocs.io/en/latest/overview.html>>
- [Joo] Joomla!. <<https://www.joomla.org/>>
- [Lif] LifeRay. <<https://www.liferay.com/downloads>>
- [LR] Liferay. <<https://www.liferay.com/digital-experience-platform>>
- [Luc] Lucid. <<http://www.lucid-desktop.org/documentation/en/dev/user/install/instructions/>>
- [MInst] Moodle Installation quick guide. <https://docs.moodle.org/31/en/Installation_quick_guide>
- [Moo] Moodle. <<https://moodle.org/?lang=es>>
- [Mshare] CMS Market Share. <<http://www.opensourcecms.com/general/cms-marketshare.php>>
- [OBr] OpenBravoERP. <<http://www.openbravo.com/es/product-download/>>
<<https://sourceforge.net/projects/openbravo/files/02-openbravo-sources/3.OPR16Q2.1/>>
- [Obs] Observium. <<http://www.observium.org/>>
- [Odo] ERP Oddo. <https://www.odoo.com/es_ES/page/download>
- [OE] OneEye. <<https://wiki.oneeye-project.org/0.9:demo>>
- [OIns] Observium Debian/Ubuntu Installation. <http://www.observium.org/docs/install_debian/>
- [ONMS] The Network Management Plataform. <<http://www.opennms.org/en>>
- [ONMSI] OpenNMS Install. <<http://www.opennms.org/en/install-debian>>
- [Ono] OnOS. <<http://onosproject.org/>>
- [OS.js] OS.js. <<http://os.js.org/>>
- [OTRS] Open-source Ticket Request System. <<https://www.otrs.com/descargue-el-software-de-mesa-de-ayuda-de-codigo-abierto-de-otrs-y-sus-funciones-gratuitas/?lang=es>>
- [Pan] PandoraFMS. <http://wiki.pandorafms.com/index.php?title=Pandora:Documentation_en:Installing>

[Piw] Piwik, Open Analytics Platform. <<https://piwik.org/docs/installation/#the-5-minut-piwik-installation>>

[Plo] The Ultimate Enterprise CMS. <<https://plone.org/>>

[PPi] ProjectPier. <<https://sourceforge.net/projects/projectpier/files/projectpier/>>

[Pro] Promox Virtual Environment.
<http://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_Jessie>

[PrS] Prestashop, ecommerce CMS. <<https://www.prestashop.com/en/system-requirements>>

[PyI] Pydio Community. <<https://pydio.com/en/community>>

[Rac] RacjTables. <<https://sourceforge.net/projects/racktables/>>

[Red] RedMine. <<http://www.redmine.org/>>

[RedI] HowTo Install Redmine on Ubuntu step by step.
<http://www.redmine.org/projects/redmine/wiki/HowTo_Install_Redmine_on_Ubuntu_step_by_step>

[Sen] Sensu. <<https://sensuapp.org/downloads>>

[Shi] Shinken Monitoring System. <<http://www.shinken-monitoring.org/>>

[Spl] Splunk. <https://www.splunk.com/en_us/download/splunk-enterprise.html>

[Sug] SugarCRM CE. <<http://www.sugarcrm.com/download>>

[Syn] Quick Start Guide: Synthesize. <<http://graphiteapp.org/quick-start-guides/synthesize.html>>

[Syn] SyncAny. <<https://www.syncany.org/>>

[Tra] Trac. <<https://trac.edgewall.org/>>

[VRoUt] **González Ramírez, M.; Gascó Gascó, J.; Llopis Taverner, J.** (2015). "Razones y riesgos del *outsourcing* de sistemas de información en las grandes empresas españolas". *Revista Europea de Dirección y Economía de la Empresa* (núm. 24).
<<http://www.elsevier.es/es-revista-revista-europea-direccion-economia-empresa-346-articulo-razones-riesgos-del-outsourcing-sistemas-S1019683815000177>>

[WP] WordPress. <<https://es.wordpress.org/>>

[Zen] Zenoss. <<http://www.zenoss.org/>>

[Zim] Zimbra. <<https://www.zimbra.com/downloads/zimbra-collaboration-open-source/>>