

## Unidad 5: Gestores de contenidos

El gestor de contenidos es una herramienta que puede trabajar de forma autónoma o, y esta es la manera más habitual, como herramienta web. Para poder trabajar con el gestor de contenidos como herramienta web, necesitará preparar el sistema para trabajar como un servidor web completo. Esto implica instalar un servidor de páginas web dinámicas (Apache y PHP) o servidores de aplicaciones (Tomcat (JSP) y un servidor de base de datos (MySQL, Postgres).

El trabajo de administrador de un gestor de contenidos no termina con la instalación y la administración del sistema. Una parte fundamental del trabajo de cualquier administrador informático es la documentación del trabajo en el momento en que se están llevando a cabo. Instalarlos un sistema wiki en su servidor para poder crear la documentación paralelamente al trabajo hecho sobre el servidor y tenerla ordenada y accesible.

### 1. Instalación de un gestor de contenidos

*Un gestor de contenidos (en inglés Content Management System o CMS) es una aplicación informática utilizada para crear, editar, gestionar, buscar y publicar diferentes tipos de contenido multimedia.*

Los gestores de contenidos se utilizan a menudo para publicar documentación específica, tales como artículos de noticias o manuales técnicos, aunque en general se pueden utilizar para crear cualquier tipo de portal web. Son capaces de gestionar contenido en diferentes formatos, como texto, imágenes, vídeos, sonidos, animaciones y cualquier otro contenido web.

Un gestor de contenidos puede tener, entre otras, alguna de las características siguientes:

- Identificar diferentes tipos de usuario y sus roles (lo que podrán ver y hacer estos usuarios).
- Crear contenidos de forma rápida y sencilla.
- Separar el contenido de su presentación. Esto permite modificar fácilmente la apariencia, sin tener que cambiar los contenidos. Además, nos permite aprovechar los contenidos para diferentes usos (por ejemplo para visualizar una misma web desde un ordenador de mesa o desde un teléfono móvil).

Hay diferentes tipos de gestores de contenidos en función de cuál es el resultado final que se quiere obtener (un centro de recursos de una empresa, la documentación de un proyecto ...). Nos centraremos en los gestores de contenidos enfocados a obtener como resultado un portal web.

Los gestores de contenidos web surgieron a mediados del noventa. Suelen estar implantados como aplicaciones web: una web que sirve para configurar y crear contenidos que da como resultado otra web. Son herramientas muy sencillas de usar, por lo que usuarios sin ningún tipo de conocimiento de programación son capaces de crear

contenido con facilidad.

Aunque los gestores de contenidos en línea (online) son los más populares, también hay gestores de contenidos fuera de línea (offline) e híbridos que combinan las dos aproximaciones.

Los gestores suelen tener los contenidos almacenados dentro de bases de datos. Posteriormente, a esta información se le dará un formato para mantener un aspecto homogéneo dentro de la web.

### 1.1. Selección de un gestor de contenidos

En el mercado hay una multitud de gestores de contenidos, tanto de código abierto y gratuitos, como de software comercial y privativo. Esta es una pequeña lista de los gestores de contenidos de código abierto más populares:

- **Drupal** (<http://drupal.org>) tiene todas las funcionalidades básicas de un gestor de contenidos y, además, permite añadir foros, encuestas, votaciones, noticias ... Está implantado en PHP y SQL.
- **WordPress** (<http://wordpress.org>) es un gestor de contenidos muy popular enfocado a la creación de blogs.
- **Joomla** ([www.joomla.org](http://www.joomla.org)) es uno de los gestores de contenidos más popular y reconocido, creado con PHP, MySQL y con licencia GPL. Una de sus características más importantes es que dispone de muchas extensiones existentes que añaden nuevas funcionalidades y características y se integran fácilmente.
- **Liferay** (<https://www.liferay.com/es>) es uno de los gestores de contenidos más robustos y escalables del mercado, creado en java , con licencias de pago para empresa y GPL en su versión libre para al comunidad. Una de sus características más importantes es que constituye un marco de trabajo para realizar todo tipo de proyectos con funcionalidades de base y que se integran fácilmente con otras plataformas y tecnologías.

Hay una infinidad de otros gestores de contenidos, tales como *Alfresco*, *Plone*, *Symphony*, *Django*, *PostNuke* ... No haremos una descripción de cada uno, pero tienen unas características similares. Los gestores de contenidos se pueden diferenciar en la forma en que se administran los contenidos, su implementación interna (lenguaje de programación y base de datos utilizada) o qué sector están orientados (empresarial, educativo ...).

Escogeremos *Liferay* como gestor de contenidos para este curso, ya que es muy fácil de instalar, hay muchas extensiones y es uno de los gestores de contenidos más robustos, por tanto, los que tienen una comunidad de desarrolladores más activa.

Las webs siguientes han sido diseñados con Liferay y sirven de ejemplo de las páginas web que podemos conseguir como resultado:

<https://www.liferay.com/es/resources?93350categoryIds=93352>

Como puede observarse, los resultados pueden ser de lo más variados y profesionales. Liferay está publicado bajo una licencia GNU Lesser General Public License, version 2.1

<http://www.gnu.org/licenses/lgpl-2.1.html>

## 1.2. Requisitos mínimos de instalación

Vamos a realizar la instalación básica de Liferay Portal 6.2, con esta instalación podremos realizar pruebas que nos ayuden a conocer la herramienta.

Antes de iniciar el manual vamos a explicar brevemente que es Liferay Portal; Liferay es un gestor de portales web y contenidos, que nos permite desde la creación de un pequeño sitio web personal, hasta la creación de varios sitios web para empresas de una forma rápida y sencilla, utilizando las funcionalidades que el propio producto nos proporciona y/o permitiéndonos desarrollar nuestro portal totalmente a medida.

Liferay Portal, proporciona dos tipos de licencias:

- Liferay Community, totalmente gratuita y descargable desde la propia página de Liferay. Tras ella hay una gran comunidad OpenSource que prueba y mejora el producto de forma continua.
- Liferay Enterprise, a la que Liferay ofrece servicios de soporte ante incidencias del producto, con la entrega continua de parches de seguridad y corrección de otro tipo de incidencias.

¡Empecemos con la instalación!

Requisitos antes de comenzar la instalación:

- Instalar JDK 1.8 y configurar la variable de entorno JAVA\_HOME
- Descargar la última versión 6.2 CE del [bundle de Liferay + Tomcat](#) aunque no sea la última incorpora la tecnología suficiente y requiere menos recursos.

Creamos la carpeta liferay, “*liferay*” en nuestro home de usuario `/home/nombreusuario/`, donde copiaremos el zip del bundle descargado.

Una vez copiado, descomprimos el zip y se creará el directorio “*liferay/liferay-portal-6.2-ce-ga6*”.

A continuación, accedemos a través de la línea de comandos a la instalación de tomcat en la ruta “*/home/nombreusuario/liferay/liferay-portal-6.2-ce-ga6/tomcat-7.0.62/bin*” y ejecutamos el comando “*startup.sh*”. Previamente le tenemos que haber dado permisos de ejecución.

De forma similar y para pararlo tenemos el comando *shutdown.sh* en el mismo directorio “*/home/nombreusuario/liferay/liferay-portal-6.2-ce-ga6/tomcat-7.0.62/bin*”

Al efectuarlo desde la línea de comandos, donde veremos como arranca el servidor. Al ser la primera vez que lo ejecutamos puede tardar varios minutos.

Durante el arranque, Liferay comprueba si tiene alguna base de datos configurada y verifica que las tablas están creadas correctamente, y en caso de no estarlo las crea de forma automática. Si no tenemos ninguna base de datos instalada, Liferay trae embebida una base de datos llamada HSQLDB (Hypersonic) con la que podremos realizar pruebas, nunca debemos usar esta base de datos en entornos productivos.

Una vez finalizado el arranque se abrirá el navegador automáticamente, y nos aparecerá la ventana de configuración básica del portal. En caso de que no se abra el navegador de forma automática, accederemos manualmente poniendo la url “<http://localhost:8080>”.

### 1.3. Preparación de la instalación sobre una base de datos Mysql

Ya hemos comentado que liferay viene con una base de datos Hypersonic de pruebas pero si queremos considerar una instalación más estable debemos optar por otro gestor de base de datos por tanto veremos como configurar nuestra instalación de Liferay con Tomcat contra una base de datos MySQL. Con esto conseguiremos hacer que la información que generemos en nuestro portal sea persistente.

#### **Requisitos previos:**

Antes de empezar a configura nuestra instalación de Liferay contra MySQL deberemos haber realizado los siguientes pasos:

1. Instalación de Liferay 6.2
2. Crear esquema de base de datos en instalación MySQL con nombre 'lportal'

Para configurar MySQL existen dos formas en Liferay, pasamos a explicar ambas:

### **Configuración a través de properties.**

La mayor parte de las configuraciones que se pueden hacer en Liferay son vía properties, uno de los principales es el “portal.properties” el cual puede consultarse en el siguiente enlace:

[portal.properties](#)

En el nos aparece un ejemplo de como configurar la conexión a base de datos.

Para poder sobrecribir los valores por defecto de este fichero, Liferay permite extenderlo. Para ello es necesario crear un fichero properties que llamaremos “portal-ext.properties” y lo podemos alojarlo en dos sitios:

- {LIFERAY\_HOME}, en el ejemplo usaremos “/home/nombreusuario/liferay/liferay-portal-6.2-ce-ga6”
- {LIFERAY\_HOME}/tomcat-7.0.62/webapps/ROOT/WEB-INF/classes

En el portal-ext.properties añadimos la configuración de nuestra base de datos, por ejemplo:

```
#Configuración del Driver para la BD MySQL
jdbc.default.driverClassName=com.mysql.jdbc.Driver
#Ruta de conexión donde indicamos el esquema de BD donde se
crearan las tablas de Liferay
jdbc.default.url=jdbc:mysql://localhost/lportal?
useUnicode=true&characterEncoding=UTF-8&useFastDateParsing=false
#Usuario y password que usaremos (debe tener permisos de creación
y edición de tablas)
jdbc.default.username=liferay
jdbc.default.password=liferay
```

Una vez arranquemos nuestro Liferay con esta configuración podremos comprobar que las tablas se han creado en nuestro esquema.

### **Configuración a través de la pantalla de inicio.**

Cuando arrancamos nuestro Liferay por primera vez sin ninguna configuración de base de datos, se mostrará una pantalla de configuración principal, en la que podremos configurar nuestra base de datos.

Ademas de los datos de configuración básica de nuestro portal, nos permite cambiar la conexión por defecto de base de datos:



## Portal

Nombre del portlet

Por ejemplo Liferay.

Lenguaje por defecto

[cambiar](#)

☐ Agregar datos de muestra

## Usuario Administrador

Nombre

Apellido

Correo electrónico (Requerido)

## Base de datos

### Base de datos por defecto (Hypersonic)

Esta base de datos es útil para hacer demos y desarrollo, pero no se recomienda su uso en producción. [\(cambiar\)](#)

[Finalizar Configuración](#)

Seleccionamos en el combo la opción MySQL.

## Base de datos

---

[« Usar la base de datos por defecto](#)

Tipo de la base de datos

DB2 ▼

DB2

Derby

Hypersonic

Ingres

MySQL

Oracle

PostgreSQL

SQLServer

Sybase

BC (Requerido)

org.hsqldb.jdbcDriver

Nombre de usuario

sa

Contraseña

Finalizar Configuración

Configuramos los datos de conexión con nuestra base de datos, esquema, usuario y contraseña.

## Base de datos

---

[« Usar la base de datos por defecto](#)

Tipo de la base de datos

MySQL ▼

URL de JDBC (Requerido)

jdbc:mysql://localhost/lportal?use

Nombre de clase del driver de JDBC (Requerido)

com.mysql.jdbc.Driver

Nombre de usuario

liferay

Contraseña

\*\*\*\*\*

Finalizar Configuración

Por último, damos a finalizar configuración y tras unos minutos, podremos comprobar que las tablas se han creado en nuestra base de datos.

Si vamos a nuestra instalación Liferay “{LIFERAY\_HOME}/” comprobaremos que se ha generado un fichero con nombre “portal-setup-wizard.properties” en el que se han incluido 4 entradas con la configuración de la base de datos.



## Configuración general

La configuración en general para adaptarlo a nuestras necesidades se hace a través de un archivo de configuración o propiedades que debe ser sobrescrito para adaptarlo.

La documentación de todos sus parámetros se puede encontrar en su referencia documental

<https://docs.liferay.com/portal/6.2/propertiesdoc/portal.properties.html>

Uno de los archivos que más información nos proporciona de las capacidades de Liferay es el archivo portal.properties que está incluido en el portal-impl.jar de Liferay. A partir de ese archivo podemos crear uno personalizado en la \$HOME de Liferay y configurar cosas como el pool de conexiones con la base de datos, los índices de lucene, las cachés, el jackrabbit, el ldap, el cas, ntml....

Se puede extraer de esta manera:

```
cd $LIFERAY_HOME
jar -xvf ./tomcat-6.0.18/webapps/ROOT/WEB-INF/lib/portal-impl.jar
portal.properties
mv portal.properties portal-ext.properties
```

Os dejo un pequeño extracto:

```
##
## JDBC
##

#
# Set the JNDI name to lookup the JDBC data source. If none is
set,
# then the portal will attempt to create the JDBC data source
based on the
# properties prefixed with "jdbc.default.".
#
#jdbc.default.jndi.name=jdbc/LiferayPool

#
# Set the properties used to create the JDBC data source.
These properties
# will only be read if the property "jdbc.default.jndi.name"
is not set.
#
# The default settings are configured for an in-memory
database called
# Hypersonic that is not recommended for production use.
Please change the
```

```
# properties to use another database.
#
# Add dynamic-data-source-spring.xml to the property
"spring.configs" to
# configure the portal to use one database cluster for read
calls and
# another database cluster for write calls. The convention is
to create a
# set of properties prefixed with "jdbc.read." to handle read
calls and
# another set of properties prefixed with "jdbc.write." to
handle write
# calls. These data sources can also be created via JNDI by
setting the
# properties "jdbc.read.jndi.name" and "jdbc.write.jndi.name".
#

#
# DB2
#
#jdbc.default.driverClassName=com.ibm.db2.jcc.DB2Driver
#jdbc.default.url=jdbc:db2:lportal
#jdbc.default.username=db2admin
#jdbc.default.password=lportal

#
# Derby
#

#jdbc.default.driverClassName=org.apache.derby.jdbc.EmbeddedDriver
#jdbc.default.url=jdbc:derby:lportal
#jdbc.default.username=
#jdbc.default.password=

#
# Hypersonic
#
jdbc.default.driverClassName=org.hsqldb.jdbcDriver
jdbc.default.url=jdbc:hsqldb:${liferay.home}/data/hsqldb/lportal
jdbc.default.username=sa
jdbc.default.password=

#
# MySQL
#
#jdbc.default.driverClassName=com.mysql.jdbc.Driver
#jdbc.default.url=jdbc:mysql://localhost/lportal?
useUnicode=true&characterEncoding=UTF-8&useFastDateParsing=false
```

```
#jdbc.default.username=
#jdbc.default.password=

#
# Oracle
#
#jdbc.default.driverClassName=oracle.jdbc.driver.OracleDriver
#jdbc.default.url=jdbc:oracle:thin:@localhost:1521:xe
#jdbc.default.username=lportal
#jdbc.default.password=lportal

#
#
# PostgreSQL

#
# Liferay uses C3PO by default for connection pooling. The
data source
# factory can be configured to use JNDI or another pooling
implementation by
# modifying infrastructure-spring.xml. See
#
http://www.mchange.com/projects/c3p0/index.html#configuration for
a list
# of additional fields used by C3PO for configuring the
database connection.
#
jdbc.default.acquireIncrement=5
jdbc.default.maxPoolSize=100
jdbc.default.minPoolSize=10
jdbc.default.numHelperThreads=10
```

## 1.5 Guía de utilización de Liferay Portal 6.2

Para empezar con cualquier gestor de contenido lo primero es entender lo que nos ofrece de base para saber como podemos reutilizarlo. Hoy en día nos permiten alcanzar muchas posibilidades únicamente adaptando su funcionalidades a nuestras necesidades pero los mejores gestores además de esto nos permiten tener herramientas para adaptar o ampliar funcionalidades donde ellos no las han alcanzado. Por ello el primer paso es practicar con lo que nos permiten hacer directamente en el producto. En documento adjunto a esta unidad mostramos una guía donde podernos familiarizar con el uso de liferay