



**eBookSync**

MANUAL DE INSTALACIÓN



## **1.- Introducción**

Este documento describe la instalación del sistema Papyre. El sistema consta de dos aplicaciones, una aplicación cliente y otra servidor.

La aplicación servidor se encarga de interactuar con ILIAS, manteniendo la información de cada alumno (cursos de los que está matriculado) y de los cursos (cuando se actualizó el curso por última vez y que recursos tiene cada curso). El objetivo es simular un servidor proxy de forma que cuando un alumno se conecte al sistema, este comprobará si el curso ha sido actualizado antes de cierto tiempo configurable, de ser así no hará nada, en caso contrario descargará la nueva información del curso. Si ahora se conecta otro alumno con el mismo curso que el anterior, no hará falta hacer conexiones con ILIAS puesto que la información que contiene nuestro sistema está actualizada.

## **2.- Software necesario**

### ***2.1.- Mysql***

Es el software encargado de la gestión de la base de datos. Lo primero será crear las bases de datos necesarias. Son dos

- eBookServer → es la base de datos que maneja la aplicación servidor. Su estructura se encuentra en el fichero bdPapyre.sql.
- Webdav → es la base de datos necesaria para webdav. Su estructura se encuentra en el fichero webdav.sql

Se deberá crear el usuario eBookServer en la base de datos.

### ***2.2.- Apache***

Para el servidor web se usará Apache. A parte de este, se deberá tener instalado php5, el módulo para trabajar con php5-mysql y el módulo php5-apache.

## 2.3.- eBookServer

Es la aplicación Java que hará de intermediario entre nuestro servidor e Ilias. Se encargará de comprobar los cursos de un alumno y descargar en el servidor los ficheros correspondientes.

Para poder ejecutar la aplicación se deben dar los siguientes pasos:

- Copiar al servidor el eBookServer\_v2.0.jar.
- En esa carpeta, copiar la carpeta lib que contiene las dependencias necesarias para la aplicación.
- Copiar al servidor el fichero configEbookServer.xml. Editarlo y hacer las modificaciones necesarias en cada campo para que el sistema esté bien configurado.

A continuación se muestra un ejemplo del fichero configEbookServer.xml

**<config>**

**<host>dv.ujaen.es</host>** Host al que se conectará.

**<connectHost><https://dv.ujaen.es></connectHost>** Protocolo de conexión al host.

**<onlyPath>/docencia/webdav.php/docencia/ref\_</onlyPath>** Path de conexión a Ilias, este se concatenará al connectHost.

**<loginDB>login</loginDB>** Login de la base de datos eBookServer.

**<passwordDB>password</passwordDB>** Password de la base de datos eBookServer.

**<driverDB>com.mysql.jdbc.Driver</driverDB>** Driver con el que se conectará desde la aplicación Java a la base de datos.

**<dbToConnect>jdbc:mysql://localhost/eBookServer</dbToConnect>** Base de datos a la que se conectará la aplicación.

**<endPointSOAP><http://dv.ujaen.es/docencia/webservice/soap/server.php></endPointSOAP>** Es el endpoint de conexión por SOAP en Ilias.

**<clientSOAP>docencia</clientSOAP>** Cliente con el que se accede por SOAP.

`<timeToReload>1.0</timeToReload>` Es el tiempo fijado para que el curso sea actualizado o no. Aquí el 1.0 significa una hora (debe ser escrito en ese formato). Si por ejemplo, queremos que se actualizara cada dos horas, escribiríamos el parámetro como "2.0".

`<pathDownloadFiles>/var/www/recursos</pathDownloadFiles>` Es la ruta donde se descargarán los ficheros en el servidor.

`<debug>>true</debug>` Parametro para hacer el debuggin. Si es true, en la antalla aparecerán algunos mensajes sobre el proceso de actualización de los cursos. Si se fija como false, no aparecerán los mensajes de depuración.

`<completeHost>https://dv.ujaen.es/docencia/webdav.php/docencia</completeHost>` Es la ruta completa de acceso a Ilias por webdav.

`</config>`

Si tenemos la aplicación en nuestro home, un ejemplo de llamada sería: `java -jar /home/usuario/eBookServer.jar usuarioIlias passwordIlias /home/usuario/rutaAbsolutaFicheroConfiguracion.xml usuarioIlias`.

La primera vez que se ejecuta, nos dirá que el certificado digital de Ilias no está instalado. Esto significa que nuestra máquina virtual de java no tiene el certificado digital de dv.ujaen.es. El programa nos pedirá una ruta (absoluta) donde almacenar el certificado.

Nos generará un fichero llamada "jssecacerts". Debemos cambiarle el nombre a "cacerts". Ahora lo debemos poner en la siguiente carpeta "/usr/lib/jvm/java-X/jre/lib/security (siendo java-x la versión de la máquina virtual de java que está siendo utilizada). De esta forma ya tenemos el certificado instalado y podremos ejecutar la aplicación.

La próxima vez que se ejecute el programa, no nos dirá nada del certificado y procederá a realizar la actualización de cursos.

## ***2.4.- ebookClient***

El servidor de webdav está basado en las clases desarrolladas por Keyphrene:

(<http://blog.keyphrene.com/keyphrene/index.php/2006/09/16/9-how-to-install-a-webdav-server-in-php>)

Se ha partido de esas clases y se han realizado los cambios mínimos necesarios para incorporarlo al sistema. Cabe destacar que el sistema es un prototipo por lo que hay funcionalidades no implementadas del todo.

Esta debe ser copiada en /var/www/ebook. A continuación se describe el contenido de la carpeta y para que sirve cada uno de los ficheros:

- ebook/
  - authenticate.php → script PHP que se encarga de autenticar al usuario y de llamar al programa servidor. Aquí se define la ruta del programa servidor y el tiempo que debe pasar entre acceso y acceso para que el programa servidor sea llamado.
  - Parámetros a modificar.
    - require\_once : en todos los require\_once se debe escribir la ruta completa de los scripts a incluir.  
Ejemplo: `require_once "/var/www/ebook/inc/HTTP/WebDAV/Server/DB.php";`
    - Conexión a la base de datos.  
Ejemplo: `$db = new DB("localhost","eBookServer","eBookServer","pass_eBookServer");`
    - La llamada al exec ejecuta el programa servidor, la llamada se hace de la siguiente forma:  
`exec("java -jar rutaJAReBookServer_v2.0.jar usuarioIlias passwordIlias ficheroLog usuarioIlias")`  
Ejemplo: `exec("java -jar /home/usuario/eBookServer_v2.0.jar ".  
$_SERVER['PHP_AUTH_USER']." ".$_SERVER['PHP_AUTH_PW']."  
/home/usuario/configEbookServer.xml ".$_SERVER['PHP_AUTH_USER']."  
>> /var/www/ebook/log/logExecServer.txt &");`  
El fichero logExecServer.txt debe existir y tener los permisos correctos.
- index.php → es el script de inicio de la aplicación. Este se encarga de llamar al script autentica para que el usuario introduzca su nombre de usuario y contraseña. Si el usuario no es válido lo indica mediante un mensaje, en caso de que si lo sea, monta el sistema de ficheros. Hay algunos parámetros de configuración que deben ser revisados si se cambia la ubicación de las aplicaciones, bases de datos, etc...  
Parámetros a modificar:
  - `$DBUSER = "usuario_base_datos";`
  - `$DBPWD = "password_base_datos";`
  - `$DBHOST = "host";`
  - `$DB_WEBDAV = "nombre_db_webdav";`
  - `$server->db_user = "usuario_base_datos";`
  - `$server->db_passwd = "password_base_datos";`
  - `$server->ServeRequest("ruta_carpeta_alamcenar_archivos");`
  - *la variable require\_once con la ruta correcta.*
- ebook/inc:
  - ebook/inc/HTTP
    - ebook/inc/HTTP/WebDAV/server

- Server.php → clase base que monta el servidor de webdav. Si cambiamos el host donde se está ejecutando (por ejemplo en caso de hacer una redirección), modificar la asignación de la variable \$base\_uri.
  - Si hacemos redirección, se debe hacer la siguiente asignación:
    - *\$this->base\_uri = "http://xx.yy.es/ebook/index.php";*
  - En caso de que no se haga redirección, la asignación queda:
    - *\$this->base\_uri = \$uri;*
- ebook/inc/HTTP/WebDAV/Tools → herramienta auxiliares de la aplicación.
- ebook/inc/HTTP/WebDAV/Server/:
  - DB.php → script PHP con funciones para el manejo de la base de datos.
  - Filesystem.php → es la clase que se encarga de montar el sistema de ficheros y mostrar a cada usuario sólo los cursos que puede ver tanto por acceso web como por acceso webdav.
- ebook/log → en esta carpeta se almacenan los ficheros de log. Los ficheros de log son: aut.log, bd.log, logExecServer.txt, webdavlog.txt

Aspectos a tener en cuenta con respecto a la aplicación cliente:

- El propietario de estos ficheros y carpetas debe ser el usuario www-data puesto que se va a acceder vía web. Además deben poseer los permisos necesarios para que puedan ser accedidos y ejecutados.
- Hay que tener en cuenta el acceso los permisos de cada carpeta y fichero.
- Para la descarga de los ficheros, se debe crear la carpeta /var/www/recursos (es la ruta por defecto, se puede configurar en el fichero configEbookServer.xml de la aplicación servidor, etiqueta: <pathDownloadFiles>. Además hay que cambiarlo también en la aplicación web, en el fichero index.php modificar la línea:
 

```
$server->ServeRequest("/var/www/recursos/");
```

 poniendo la carpeta que deseemos.