



## **Manual de instalación de eXeLearning Online 2.8.1 y configuración para integración con Moodle**

Versión	0.9	20/06/2023	Pablo Amaya
	1	21/06/2023	Pablo Amaya
	1.1	26/06/2023	Pablo Amaya
	1.2	29/06/2023	Pablo Amaya
	1.3	09/01/2024	Pablo Amaya
	1.4	11/12/2024	Ignacio Gros
	1.5	12/01/2024	Pablo Amaya

## Sumario

Manual de instalación de eXeLearning Online 2.8.1 y configuración para integración con Moodle.	1
1 Introducción.....	3
2 Instalación Debian.....	3
2.1 Requisitos mínimos.....	3
2.2 Instalación de Apache.....	3
2.4 Instalación de Python 2.7.15.....	4
2.5 Configuración hosts y Apache.....	4
2.6 Despliegue de eXeLearning 2.8.1.....	5
2.7 Configuración de la integración con la extensión de Moodle.....	6
3 Instalación Ubuntu - Docker.....	6
3.1 Descarga e instalación de Docker.....	6
3.2 Despliegue de eXeLearning.....	7
3.3 Proxy inverso.....	8
3.4 Configuración de la integración con la extensión de Moodle.....	8
4 Instalación y configuración de las extensiones eXeLearning (SCORM y Sitio Web) para Moodle.	9
5 Consideraciones importantes.....	9
5.1 Redirección del servicio HTTP a HTTPS.....	9
5.2 Tamaño máximo de archivos.....	10
6 Personalización de estilos e interfaz.....	11
6.1 Instalación de estilos personalizados y estilo por defecto.....	11
6.2 Personalización de iconos e imagen general.....	11

# 1 Introducción

El presente manual contempla la instalación de eXeLearning Online versión 2.8.1 en dos sistemas distintos:

1. Usando Debian 10 e instalando y configurando paquetes de forma nativa
2. Usando Ubuntu 16 en contenedores con Docker

El entorno de eXeLearning se basa en la instalación en un servidor de:

- Apache (con sus módulos correspondientes)
- PHP (con sus módulos correspondientes)
- Python (con sus librerías correspondientes)

Para Moodle se proporciona una extensión que será instalada desde el administrador de extensiones.

## 2 Instalación Debian

### 2.1 Requisitos mínimos

Los requisitos mínimos/recomendados para la instalación son los siguientes:

- Sistema operativo Linux (Debian 10.12)
  - <https://mirror.accum.se/cdimage/archive/10.12.0/amd64/iso-dvd/>
- Procesador de 64 bits con soporte SLAT.
- 4GB de memoria RAM.
- La virtualización por hardware debe estar activada en la BIOS.
- 10GB de espacio libre en el disco duro.

### 2.2 Instalación de Apache

Instalamos Apache:

```
apt install apache2
```

Los módulos instalados por defecto son:

```
mpm_event mime dir autoindex reqtimeout access_compat setenvif deflate filter negotiation  
authz_host alias status env authz_user auth_basic authn_file authn_core authz_core
```

Deshabilitamos mpm\_event y habilitamos otros módulos:

```
a2dismod mpm_event  
a2enmod headers mpm_prefork proxy proxy_http ssl socache_shmcb
```

El listado final de módulos habilitados son:

access_compat	autoindex
alias	deflate
auth_basic	dir
authn_core	env
authn_file	filter
authz_core	headers
authz_host	mime
authz_user	mpm_prefork

```
negotiation      setenvif
proxy            socache_shmcb
proxy_http       ssl
reqtimeout       status
```

## Reiniciamos Apache

```
systemctl restart apache2
```

## 2.4 Instalación de Python 2.7.15

Instalamos los paquetes previos:

```
apt install build-essential
apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev
libgdbm-dev libc6-dev libbz2-dev
```

Instalamos Python desde los fuentes:

```
cd /usr/src
wget https://www.python.org/ftp/python/2.7.15/Python-2.7.15.tgz
tar xzf Python-2.7.15.tgz
cd Python-2.7.15
./configure --enable-optimizations
make altinstall
```

Instalamos los paquetes Python necesarios para eXeLearning con apt y con pip:

```
apt-get -y install python2.7-setuptools
apt-get -y install python2.7-zope.interface
apt-get -y install python2.7-lxml
apt-get -y install mimetex
apt-get -y install python2.7-git
apt-get -y install python-pil
apt-get -y install python-chardet
apt-get -y install python-feedparser
apt-get -yq install python-pip python-dev libxml2-dev libxmlsec1-dev swig
apt-get -yq install python-m2crypto python-isodate python-defusedxml
apt-get -yq install python-bs4 python-oauth2client python-googleapi
apt-get -yq install python-passlib
apt-get -yq install fuseext2
apt-get -yq install python-suds python-requests-oauthlib python-dateutil
apt-get -yq install python-webassets

pip install dm.xmlsec.binding==1.3.2
pip install cssmin
pip install pyjwt
```

## 2.5 Configuración hosts y Apache

Añadimos la siguiente línea al fichero /etc/hosts

```
127.0.0.1 app
```

Vamos a configurar Apache solo para un acceso por el puerto 443. Es recomendable no permitir el acceso por el puerto 80 o redirigir todas las peticiones del puerto 80 al 443. Se proporciona un fichero de configuración de Virtualhost de ejemplo usando un proxy inverso para que Apache se comunique con el servidor web Python que usa eXeLearning. En este caso, habrá que modificar los certificados SSL por los que tenga el servidor instalados y el nombre del servidor (no es recomendable usar certificados auto firmados porque Python los rechazaría).

```
/etc/apache2/sites-available/default-ssl.conf

<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
    SSLCertificateKeyFile  /etc/ssl/private/ssl-cert-snakeoil.key

    ServerName servername:443
    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>

    ProxyRequests Off
    ProxyPreserveHost On
    ProxyPass              / http://app:51235/
    ProxyPassReverse       / http://app:51235/

    ErrorLog /var/log/apache2/error.log

    LogLevel warn
    CustomLog /var/log/apache2/access.log combined
    ServerSignature On

    <Location />
        Allow from all
    </Location>
</VirtualHost>
```

## Habilitamos el VirtualHost con SSL y recargamos Apache

```
a2ensite default-ssl.conf
systemctl reload apache2
```

## 2.6 Despliegue de eXeLearning 2.8.1

El código puede ser descargado vía SSH, ZIP desde la Web o vía comando, siendo este último el proceso aquí explicado.

Como el repositorio es privado, al hacerlo vía comando con git, hay que generar un token de autenticación ([explicado en la documentación oficial de GitHub](#)). En nuestro caso podemos seguir los pasos del 1 al 10 descritos [aquí](#)

Una vez que tenemos el **token**, accedemos al directorio donde queremos desplegar y clonamos el repositorio

```
cd /var/www/html

git clone https://github.com/exelearning/iteexe_online.git

**** Nos pedirá nuestro nombre de usuario y como contraseña ponemos el token

mv iteexe_online/ iteexe/
```

## Crearemos un script para lanzar la aplicación:

```
nano run.sh

#!/bin/sh
cd /var/www/html/iteexe/
rm exe/config/exe.pid
umask 0
```

```
exe/exe --standalone --server
```

Tras esto podemos crear un servicio systemctl personalizado para la instancia

## 2.7 Configuración de la integración con la extensión de Moodle

En el fichero `./iteexe/exe/publish.conf` configuramos las direcciones y los puntos de acceso a los servicios web. En la versión actual, se indican que algunos parámetros ya no son necesarios de configurar, porque usando el JWT el sistema analiza las URL de la peticiones Moodle y se autoconfigura. Esto permite **usar una instancia eXeLearning con N instancias Moodle**.

A continuación marcamos en negrita los parámetros a configurar.

```
[info]
# Platform label name
name = 'Moodle'
    → Nombre del sistema
[url]
# Package repository home page
home = 'https://server/instancial'
    → Instancia a conectar. Este parámetro ya no es necesario configurar en el
    caso de Moodle
# Services
get_ode = 'https://server/instancial/mod/exelearning/get_ode.php'
    → Este parámetro ya no es necesario configurar en el caso de Moodle
set_ode = 'https://server/instancial/mod/exelearning/set_ode.php'
    → Este parámetro ya no es necesario configurar en el caso de Moodle
[config]
# Open ode_uri in browser when publishing
# - '0' => disabled
# - '1' => enabled
publish_redirect_enabled = '1'
# In current or new window
# - '0' => current window
# - '1' => new window
publish_redirect_blank = '0'
# Integration logs (set_ode/get_ode)
# - '0' => Disabled logs
# - '1' => Enabled logs
enabled_logs = '1'
# Use Integration secret key in jwt
# - '0' => Disabled
# - '1' => Enabled
enabled_jwt = '1'
    → Habilitamos el sistema JWT
jwt_secret_key = 'claveXXXXXX'
    → Clave compartida entre servidores. Esta clave habrá que incorporarla a los
    servidores con los que se conecte la instancia eXeLearning
jwt_secret_hash = 'HS256'
```

## 3 Instalación Ubuntu - Docker

Sobre una instalación de Ubuntu 16 (<https://releases.ubuntu.com/xenial/ubuntu-16.04.7-server-amd64.iso>) comenzamos instalando Docker.

### 3.1 Descarga e instalación de Docker

Instalamos los paquetes previos y configuramos APT.

```
sudo apt-get update
```

```
sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo apt-key fingerprint 0EBFCD88
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Instalamos Docker y añadimos a un usuario del grupo `sudoers` para que pueda ejecutarlo

```
sudo apt-get update
sudo apt-get install docker-ce
sudo usermod -aG docker usuario
```

Instalación de Docker Compose y comprobar que la versión es la 1.21.2 recién descargada

```
sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
docker-compose version
```

### 3.2 Despliegue de eXeLearning

Suponemos la instalación dentro de `/opt`

```
sudo mkdir /opt/exelearning
cd /opt/exelearning/
```

El código puede ser descargado vía SSH, ZIP desde la Web o vía comando, siendo este último el proceso aquí explicado.

Como el repositorio es privado, al hacerlo vía comando con git, hay que generar un token de autenticación ([explicado en la documentación oficial de GitHub](#)). En nuestro caso podemos seguir los pasos del 1 al 10 descritos [aquí](#)

Una vez que tenemos el **token**, accedemos al directorio donde queremos desplegar y clonamos el repositorio

```
git clone https://github.com/exelearning/iteexe_online.git

**** Nos pedirá nuestro nombre de usuario y como contraseña ponemos el token

mv iteexe_online/ iteexe/

cd iteexe
```

Dentro de la carpeta `images` **tenemos tres contenedores configurados** para montar Apache, el IDP y eXeLearning 2.8.1. Los ficheros Docker ya están configurados con todos los requisitos. Es importante destacar que el contenedor IDP solo se usa si usáramos Google Authenticator y SimpleSAMLphp, por lo que no es necesario levantarlo en el caso de usar JWT y Moodle.

Al igual que en la instalación Debian, los certificados SSL (no se recomiendan autofirmados) hay que indicarlos en el fichero `docker-compose.yml`

Habría que añadir la línea que se marca en negrita y copiar los certificados SSL al directorio `images/apache2/certs`

```
sudo nano /opt/exelearning/iteexe/docker-compose.yml
```

```
volumes:
  - ./images/apache2/sites:/etc/apache2/sites-enabled
  - ./images/apache2/htpasswd:/etc/apache2/htpasswd
  - ./images/apache2/log:/var/log/apache2
  - ./images/apache2/certs:/certs
```

Por último lanzamos los contenedores con:

```
sudo docker-compose up -d
```

```
pabloamaya@exelearning:/opt/exelearning/iteexe$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
62c3eddc75a2   iteexe_front   "/bin/sh -c /run.sh"    About a minute Up About a minute 0.0.0.0:80->80/tcp, :
e428ef797863   iteexe_app     "images/exe/run.sh"     About a minute Up About a minute 0.0.0.0:8180->80/tcp,
399fe627376a   iteexe_idp     "/bin/sh -c /run.sh"    About a minute Up About a minute 0.0.0.0:8180->80/tcp,
```

Podemos comprobar que los tres contenedores están levantados con `sudo docker ps -a`

El siguiente paso sería configurar el archivo de integración eXeLearning-Moodle (`publish.conf`). Para ello habría que configurar dicho fichero siguiendo los pasos del punto 3.4 entrando en el contenedor `iteexe_app`.

```
sudo docker exec -it iteexe_app_1 /bin/bash
```

### 3.3 Proxy inverso

El acceso externo a esta aplicación debería ser a través de proxy inverso y redireccionando todo el tráfico hacia la conexión HTTPS del equipo que ejecute los contenedores Docker.

A continuación mostramos un ejemplo de configuración de un Virtualhost en Apache:

```
<VirtualHost IP:443>
    ServerName nombreservidor
    ServerAlias nombre2
    ProxyPreserveHost On
    ProxyRequests Off
    SSLProxyEngine On
    ProxyPass / https://IP2/
    ProxyPassReverse / https://IP2/

    ProxyHTMLURLMap ^/(.*)$ /$1
    SSLProxyCheckPeerCN Off
    SSLProxyCheckPeerName Off

    SSLCertificateFile /etc/cert/cert.pem
    SSLCertificateKeyFile /etc/cert/privkey.pem
    .....
</VirtualHost>
```

### 3.4 Configuración de la integración con la extensión de Moodle

Habría que entrar en el contenedor denominado **app** para modificar `publish.conf` de la misma forma que se explica en el apartado **2.7 Configuración de la integración con la extensión de Moodle**.



## 4 Instalación y configuración de las extensiones eXeLearning (SCORM y Sitio Web) para Moodle

Las extensiones las podemos descargar como fichero ZIP o con GIT desde

[https://github.com/exelearning/mod\\_exescorm](https://github.com/exelearning/mod_exescorm)


[https://github.com/exelearning/mod\\_exeweb](https://github.com/exelearning/mod_exeweb)

En caso de usar el comando git, habría que usar el token generado tal y como se explicó en **2.6 Despliegue de eXeLearning 2.8.1**

Una vez subidas e instaladas las extensiones en Moodle (hay que configurar cada extensión independientemente ), procedemos a configurarlas, siendo obligatorio configurar los dos siguientes parámetros:

- eXeLearning (SCORM)**

Configuración de la conexión con eXeLearning

URI remoto <small>exelearning   exeonlinebaseuri</small>	<input type="text" value="https://servidorex.dominio.es"/>	Valor por defecto: Vacío
Clave de firma <small>exelearning   hmakey1</small>	<input type="text" value="claveXXXXX"/>  Clave utilizada para firmar los datos enviados al servidor de eXeLearning, ( de 32 caracteres.	
- URI remoto.**  
 Dirección donde hemos desplegado eXeLearning Online  
**Clave de firma.** Clave compartida entre los servidores

## 5 Consideraciones importantes

### 5.1 Redirección del servicio HTTP a HTTPS

Es importante redirigir todo el tráfico, tanto en el servidor Moodle como en eXeLearning, del puerto HTTP a HTTPS. También sería recomendable deshabilitar el acceso por HTTP.

## 5.2 Tamaño máximo de archivos

Para la correcta integración Moodle-eXeLearning hay que configurar ciertas variables para establecer el tamaño máximo de los ficheros con los que se va a trabajar e intercambiar entre servidores.

- PHP
  - `post_max_size` y `upload_max_size`
- Apache
  - `LimitRequestBody`
- En Moodle
  - Tamaño máximo de subida de archivos (*Administración del Sitio → Seguridad → Políticas de seguridad del sitio → Tamaño máximo del archivo subido*).
  - Tamaño máximo para subida de ficheros que se configure en cada curso.
- En eXeLearning Online
  - Dentro del fichero `exe/engine/config.py` tenemos tres variables:
    - `maxUploadSizeTinyEditorMCE` → Tamaño máximo permitido para subir archivos en el editor TinyMCE
    - `maxSizeImportElp` → Tamaño máximo permitido a la hora de importar un ELP
    - `maxSizePublish` → Tamaño máximo para poder publicar desde eXeLearning a Moodle. Este valor es el que aparece como tamaño máximo de proyecto en la barra superior de eXeLearning. En este caso, para el cálculo del tamaño de proyecto no se tiene en cuenta si el proyecto tiene activada la opción de “Exportar una copia del .elp”, si se incluye un enlace al propio .elp desde TinyMCE o si se ha incluido el iDevice “Descargar el fichero fuente”. Por lo tanto, **recomendamos que el valor máximo de subida de archivos que permita el servidor Moodle sea, al menos, el doble de lo que se establezca en `maxSizePublish`.**

## 6 Personalización de estilos e interfaz

### 6.1 Instalación de estilos personalizados y estilo por defecto

Por defecto, los usuarios no podrán instalar estilos en la versión online, al igual que no podrán crear iDevices personalizados. Más adelante explicamos cómo hacer para que sí puedan. Antes, explicaremos **cómo establecer la lista de estilos disponibles**:

eXe, al arrancar, lee los estilos que se encuentran en `/exe/webui/style`, y los muestra en el menú Estilos de la aplicación.

Si queremos añadir o quitar estilos no hay más que incluir o borrar carpetas en ese directorio. Si queremos, por ejemplo, que el estilo "Docs" no esté disponible, borramos la carpeta "docs" y reiniciamos el servicio, sin más.

Es importante definir el estilo por defecto. Eso se hace en el fichero `/exe/engine/config.py` (`defaultStyle`). Por defecto es INTEF, pero podemos usar cualquiera de los que tengamos en la carpeta `/exe/webui/style`.

¿Qué hacemos al establecer un estilo por defecto?

- Aplicamos ese estilo a los contenidos nuevos.
- Lo aplicamos a los contenidos que usan un estilo que no tenemos instalado. Por ejemplo: descargamos un REA de un repositorio, y ese REA usa un estilo que nuestro eXe online no tiene. Podremos editar el contenido sin problemas, pero se abrirá con el `defaultStyle`. El usuario podrá cambiar luego el estilo, por supuesto.

Si queremos que los usuarios puedan instalar estilos, o incluso editar algunos iDevices, tendremos que personalizar la barra de herramientas de nuestro eXe. Más adelante explicamos cómo hacerlo (punto "Personalización de la barra de herramientas"). Modificando esa barra también podremos elegir el orden en el que se presentan nuestros estilos, para, por ejemplo, colocar en primer lugar nuestros estilos institucionales, y luego los demás.

### 6.2 Personalización de iconos e imagen general

Pisando los siguientes ficheros, podemos sustituir el favicon de la aplicación:

- `eXe_icon.ico`
- `/exe/webui/images/eXe_icon.ico`
- `/exe/webui/images/favicon.ico` Si pisamos este fichero, y exportamos usando un estilo que no tiene favicon propio, será este fichero el que se exporte. Es decir: se trata del favicon por defecto para las exportaciones.

Pisando la imagen `exe/webui/images/exe_header_logo.png` podemos personalizar el icono que se ve en la parte superior izquierda de la aplicación. Es un icono pequeño, de 21x20 píxeles. Debemos respetar sus dimensiones.

También podemos personalizar fácilmente las páginas que se muestran cuando eXe está cargando o importando un contenido. Podemos, por ejemplo, mostrar un logotipo institucional propio en lugar del logotipo de eXeLearning. Para eso tenemos que hacer dos cosas: incluir nuestro logotipo y hacer que se muestre en esas páginas mediante CSS. Por ejemplo:

Subimos nuestro logo a `exe/webui/images/nuestroLogo.png`, y luego añadimos unas líneas al fichero `/exe/jsui/scripts/exe_custom.css` para que sea ese logo, y no el que tiene eXe por defecto, el que se muestre. Ejemplo:

```
#loading .loading-indicator {  
    background-image:url (/images/nuestroLogo.png);  
    background-size:300px auto  
}
```

Desde ese mismo fichero CSS podemos personalizar cualquier otro elemento de la aplicación. Eso está al alcance de cualquier programador con conocimientos de HTML/CSS. Ejemplo de personalización del botón de "Finalizar" (parte superior derecha de la aplicación):

```
.file_export_finish_btn {  
    background:#28A745;  
    border-color:#28A745  
}  
.file_export_finish_btn:hover,.file_export_finish_btn:focus {  
    background:#218838;  
    border-color:#218838  
}  
.file_export_finish_btn .x-btn-inner {  
    color: #FFF  
}
```

### 6.3 Personalización de la barra de herramientas

Ya se ha comentado que la gestión de estilos e iDevices está deshabilitada por defecto. Permitir a los usuarios añadir estilos implica permitir generar contenidos con cualquier apariencia, y permitir gestionar iDevices implica que se pueden crear iDevices personalizados que no puedan ser editados por otro usuario.

En cualquier caso, si se desea habilitar esas opciones, basta con buscar la cadena "*Disabled on this version*" en el fichero `/exe/jsui/scripts/app/view/ui/eXeToolbar.js`, y descomentar esas líneas.

Pisando ese fichero también podremos añadir o quitar elementos de nuestra barra de herramientas.

Si lo que queremos es únicamente cambiar el orden de los estilos disponibles, como se ha sugerido anteriormente, es más cómodo hacerlo pisando el fichero `/exe/jsui/scripts/app/controller/Toolbar.js`; en concreto, el método `stylesRender`.

En cualquier caso, hay que tener en cuenta que pisar cualquiera de esos dos archivos puede complicar el mantenimiento de la aplicación, ya que el archivo original de eXeLearning puede cambiar de una versión a otra.

#### 6.4 Problemas habituales en algunos estilos

El módulo exeweb para Moodle requirió modificaciones en la mayoría de los estilos de eXeLearning, incluido el estilo Base, que ha servido como punto de partida para la creación de muchos estilos. El estilo DUA, que parte de un estilo desarrollado por la Junta de Andalucía y que a su vez ha servido como punto de partida para la creación de estilos DUA en otras CCAA, también tuvo que ser modificado.

**Se recomienda revisar los estilos propios.** En concreto:

- Hay que evitar que el menú del sitio web se vuelva a mostrar cuando se redimensiona la ventana. El motivo es que esa ventana se redimensiona constantemente cuando el contenido se muestra en un IFRAME en Moodle al usar el módulo exeweb. Ejemplo de cambios a realizar (los cambios corresponden al estilo DUA):  
<https://github.com/exelearning/iteexe/commit/1e87e3efcfcd7836b1c4783f245e9e1a08143cf>  
c
- Hay que corregir los estilos similares a DUA para evitar que la altura de ese mismo IFRAME en Moodle crezca indefinidamente provocando efectos visuales no deseados. Ejemplo de cambios a realizar (los cambios corresponden al estilo DUA):  
<https://github.com/exelearning/iteexe/commit/1e87e3efcfcd7836b1c4783f245e9e1a08143cf>  
c