

Guía de desarrollador

2024-09-16

Table of contents

Prefacio	5
¿A quién está dirigido?	5
¿Qué encontrarás en este manual?	5
Licencias y filosofía del proyecto	6
Tu papel en el desarrollo	6
1 Introducción	7
1.1 Resumen del plugin	7
1.2 Características clave del plugin:	7
1.3 Estructura del documento	8
2 Requisitos previos	9
2.1 Software necesario	9
2.1.1 Moodle	9
2.1.2 PHP	9
2.1.3 Servidor local o remoto	9
2.1.4 Composer	10
2.1.5 Servidor SQL	10
2.1.6 Cliente SQL	10
2.2 Conocimientos recomendados	10
2.3 Configuración de un entorno de desarrollo	11
2.3.1 Paso 1: Configura un servidor local	11
2.3.2 Paso 2: Instala Moodle	11
2.3.3 Paso 3: Instala Composer	11
2.3.4 Paso 4: Configura tu editor o IDE	11
3 Estructura del <i>plugin</i>	12
3.1 Descripción general	12
3.1.1 <code>backup/</code>	12
3.1.2 <code>classes/</code>	13
3.1.3 <code>db/</code>	13
3.1.4 <code>lang/</code>	13
3.1.5 <code>vendor/</code>	13
3.1.6 Archivos principales	14
3.2 Buenas prácticas al trabajar con la estructura	14

4	Instalación del <i>plugin</i>	15
4.1	Descarga e instalación	15
4.1.1	Paso 1: Obtener los archivos del <i>plugin</i>	15
4.1.2	Paso 2: Copiar el <i>plugin</i> al directorio adecuado	15
4.1.3	Paso 3: Comprobar los permisos de los archivos	15
4.2	Activación del <i>plugin</i> en Moodle	16
4.2.1	Paso 1: Acceder al panel de administración	16
4.2.2	Paso 2: Comprobar la instalación automática	16
4.2.3	Paso 3: Revisión de requisitos	16
4.2.4	Paso 4: Completar la instalación	16
4.3	Configuración inicial del plugin	17
4.3.1	Paso 1: Acceder a la configuración	17
4.3.2	Paso 2: Configurar la API Key de OpenAI	17
4.4	Comprobaciones finales	17
4.4.1	Paso 1: Verificar la funcionalidad	17
4.4.2	Paso 2: Habilitar la depuración (opcional)	17
4.4.3	Paso 3: Probar diferentes configuraciones	17
4.5	Actualización del <i>plugin</i>	18
4.5.1	Paso 1: Obtener la nueva versión	18
4.5.2	Paso 2: Reemplazar los archivos	18
4.5.3	Paso 3: Completar la actualización	18
4.5.4	Paso 4: Verificar compatibilidad	18
4.6	Notas adicionales	18
5	Uso del plugin	19
5.1	Introducción al flujo básico del <i>plugin</i>	19
5.2	Añadir una nueva actividad de cuestionario adaptativo	19
5.2.1	Paso 1: Acceso a la configuración del curso	19
5.2.2	Paso 2: Añadir una actividad	19
5.2.3	Paso 3: Configuración inicial	20
5.2.4	Paso 4: Guardar y procesar	20
5.3	Configuración avanzada de las preguntas	20
5.4	Publicación del cuestionario	21
5.5	Resolución del cuestionario por parte de los estudiantes	21
6	Desarrollo y personalización	22
6.1	Añadiendo funcionalidades	22
6.1.1	Añadir soporte para otro proveedor de LLM	22
6.1.2	Añadir nuevos eventos	23
6.2	Personalización de las preguntas generadas	23
6.2.1	Modificación de parámetros enviados a OpenAI	23
6.2.2	Control del formato de las preguntas	24
6.2.3	Modificación del esquema de base de datos	24

7	Solución de problemas	26
7.1	Errores comunes y sus soluciones	26
7.1.1	Error durante la instalación del plugin	26
7.1.2	El cuestionario no se genera correctamente	26
7.1.3	Problemas de rendimiento en el servidor	27
7.1.4	Conflictos con otros plugins	27
7.2	Habilitar la depuración en Moodle	28
7.3	Uso de los registros (<i>logs</i>)	28
7.4	Solución de conflictos de dependencias	29
7.5	Preguntas frecuentes	29
7.5.1	¿Por qué no puedo subir archivos grandes al plugin?	29
7.5.2	¿Qué hacer si la API de OpenAI no responde?	29
7.5.3	¿Cómo depurar errores relacionados con las preguntas generadas?	30
8	Licencias y contribuciones	31
8.1	Licencia del código fuente: GPLv3	31
8.1.1	Obligaciones de la GPLv3	31
8.2	Licencia de la documentación: CC-BY-SA	32
8.3	Contribuciones al proyecto	32
8.4	Reconocimientos	33
9	Referencias y recursos	34
9.1	<i>Moodle Developer Docs</i>	34
9.2	Repositorio del plugin	34
9.3	Documentación de OpenAI	35
9.4	Recursos sobre PHP y Composer	35
9.5	Recursos para el manejo de dependencias	35
9.6	Recursos adicionales	36
9.7	Recursos para pruebas	36

Prefacio

Este es el manual para desarrolladores del *plugin* DALEK: Digital Analysis and Learning Enhancement Kit, una herramienta diseñada para mejorar la eficiencia de los docentes a la hora de evaluar mediante cuestionarios dentro de la plataforma Moodle. Este documento tiene como objetivo ser una guía para quienes deseen entender, personalizar o extender el funcionamiento de DALEK, facilitando tanto su integración como su evolución en entornos educativos.

Este manual se ha creado con dos objetivos principales:

1. Facilitar la comprensión técnica del *plugin*
2. Promover su personalización y mejora continua

¿A quién está dirigido?

Este documento está dirigido a desarrolladores con conocimientos básicos o avanzados de programación, familiarizados con PHP y la arquitectura de Moodle. También es útil para administradores de sistemas educativos que necesiten entender el funcionamiento del plugin para mantenerlo en entornos de producción.

Si eres nuevo en el desarrollo de Moodle, no te preocupes: el manual incluye explicaciones claras (esperamos) y referencias a documentación externa que te guiarán en los conceptos fundamentales. Si ya tienes experiencia, encontrarás información detallada que te permitirá trabajar directamente en la personalización del plugin.

¿Qué encontrarás en este manual?

Este manual está organizado de la siguiente manera:

- Una descripción detallada de la estructura del plugin, incluyendo su integración con la API de Moodle y las dependencias externas que utiliza.
- Instrucciones paso a paso para instalar y configurar DALEK en tu plataforma Moodle.
- Guías para personalizar el plugin y añadir nuevas funcionalidades.
- Soluciones a problemas comunes, métodos para depurar errores y sugerencias para realizar pruebas efectivas.

- Información sobre licencias y cómo contribuir al desarrollo del plugin o usar su base para nuevos proyectos.

Licencias y filosofía del proyecto

El plugin DALEK está licenciado bajo GPLv3, lo que significa que puedes usar, modificar y redistribuir el código siempre que respetes los términos de esta licencia. Por su parte, este manual está licenciado bajo CC-BY-SA, lo que fomenta su uso, adaptación y mejora, siempre que se reconozca el trabajo original y se mantenga la misma licencia en versiones derivadas.

Esta filosofía abierta refleja los valores del proyecto: creemos que la educación debe estar al alcance de todos, y que las herramientas tecnológicas que la respaldan deben ser accesibles y adaptables.

Tu papel en el desarrollo

El desarrollo de DALEK no termina con este manual ni con su primera versión. Como desarrollador, tu contribución es clave para ampliar las posibilidades del *plugin*. Ya sea solucionando un problema, implementando una nueva funcionalidad, mejorando y ampliando la documentación o compartiendo ideas para mejorar el sistema, tu participación ayuda a mantener vivo el proyecto y a hacerlo más útil para la comunidad educativa.

Esperamos que este manual no solo sea una herramienta práctica, sino también una invitación a formar parte de un esfuerzo colaborativo para mejorar la educación.

1 Introducción

Este documento está diseñado para ser una guía técnica completa destinada a los desarrolladores interesados en trabajar con el plugin DALEK: Digital Analysis and Learning Enhancement Kit. Su objetivo es proporcionar toda la información necesaria para comprender, instalar, personalizar y extender el plugin en entornos Moodle. Además, incluye detalles sobre su arquitectura interna, los requisitos técnicos, y consejos para la solución de problemas.

El manual está pensado tanto para aquellos que desean contribuir al desarrollo del plugin como para quienes buscan integrarlo o adaptarlo a las necesidades específicas de sus plataformas educativas.

1.1 Resumen del plugin

DALEK es un plugin desarrollado para Moodle, que utiliza modelos de lenguaje avanzados (LLM, como ChatGPT) para generar cuestionarios adaptativos de manera automática. Este sistema tiene como objetivo mejorar la personalización de la evaluación educativa, creando preguntas basadas en contenidos textuales proporcionados por los docentes, como memorias, informes o apuntes en formato PDF.

La principal ventaja de DALEK es su capacidad para reducir la carga de trabajo de los docentes a la hora de diseñar actividades de evaluación y garantizar que estas estén alineadas con los contenidos. Además, fomenta una evaluación más equitativa y personalizada, ajustándose al nivel y contexto de cada estudiante.

1.2 Características clave del plugin:

- Generación automática de preguntas a partir de documentos de texto, utilizando la API de OpenAI.
- Integración completa con Moodle, respetando su estructura de permisos, roles y flujos de trabajo.
- Configuración flexible para adaptar la generación de cuestionarios a las necesidades de cada docente o institución.
- Soporte multilingüe a través del sistema de internacionalización de Moodle.

1.3 Estructura del documento

El manual está organizado en capítulos que cubren todos los aspectos del *plugin*. A continuación, se resumen los contenidos que se incluyen:

- **Requisitos previos:** Software y conocimientos necesarios para instalar y trabajar con el plugin.
- **Estructura del plugin:** Explicación detallada de la arquitectura de carpetas y archivos.
- **Instalación del plugin:** Instrucciones paso a paso para instalar y configurar DALEK en Moodle.
- **Uso del plugin:** Guía para los usuarios finales y administradores que deseen configurar actividades con el *plugin*.
- **Desarrollo y personalización:** Información para programadores que deseen ampliar o modificar la funcionalidad del *plugin*.
- **Solución de problemas:** Consejos para resolver errores comunes y utilizar herramientas de depuración.
- **Pruebas y validación:** Metodologías para asegurar el correcto funcionamiento del *plugin*.
- **Licencias y contribuciones:** Detalles sobre las licencias aplicadas y cómo colaborar con el proyecto.
- **Referencias y recursos:** Enlaces y documentación útil para aprender más sobre Moodle, OpenAI y herramientas relacionadas.

2 Requisitos previos

Antes de empezar a trabajar con el *plugin*, es importante asegurarse de cumplir con una serie de requisitos, tanto de software como de conocimientos. Este apartado detalla todo lo necesario para una instalación correcta y una buena preparación para el desarrollo.

2.1 Software necesario

Para que el *plugin* funcione correctamente y puedas desarrollarlo o modificarlo, debes contar con el siguiente software:

2.1.1 Moodle

Moodle es el entorno en el que opera el plugin. Es imprescindible que la instalación esté activa y correctamente configurada en un servidor accesible, por lo que si no es así, , descárgalo desde su [sitio oficial](#), y sigue las guías de instalación. Asegúrate de que tu servidor cumple los requisitos de la versión de Moodle.

- **Versión requerida:** Moodle 4.1 o superior.

2.1.2 PHP

PHP es el lenguaje principal en el que está desarrollado Moodle y este *plugin*. La versión 8.1 garantiza compatibilidad con las funcionalidades modernas del lenguaje y el rendimiento necesario para manejar la carga de trabajo de Moodle.

- **Versión requerida:** PHP 8.1 o superior.

2.1.3 Servidor local o remoto

Si trabajas en local para desarrollo, utiliza un servidor como XAMPP, WampServer o MAMP (según tu sistema operativo). Para entornos de producción o pruebas avanzadas, configura un servidor remoto con herramientas como Apache o Nginx.

2.1.4 Composer

Composer es una herramienta de gestión de dependencias para PHP.

Este *plugin* utiliza dependencias externas que se gestionan con Composer por lo que, si no lo tienes instalado, es recomendable que lo hagas. Basta con descargarlo desde su [sitio oficial](#), instalarlo y asegurarte de que se encuentra en el PATH de tu sistema.

2.1.5 Servidor SQL

Es otro de los requisitos necesarios para que Moodle funcione. Prácticamente cualquiera de los principales sistemas gestores de bases de datos nos vale (e.g. MariaDB, PostgreSQL). Aun así, comprueba el manual de requisitos de Moodle para instalar el que mejor se ajuste a tus necesidades.

2.1.6 Cliente SQL

Para interactuar con la base de datos desde fuera de Moodle (algo muy útil en entornos de desarrollo), viene bien tener un cliente SQL como:

- DBeaver (multiplataforma).
- HeidiSQL (Windows).
- MySQL Workbench (multiplataforma y nuestro preferido si usas MySQL o MariaDB como gestor de bases de datos).
- phpMyAdmin, si prefieres una interfaz web.

2.2 Conocimientos recomendados

Aunque no es estrictamente necesario ser un experto, tener ciertos conocimientos facilitará mucho el proceso de instalación, personalización o desarrollo del plugin.

Es importante conocer las bases del lenguaje PHP, ya que el código del plugin está escrito en este lenguaje. También es interesante tener conocimientos de cómo interactuar con API REST, ya que el plugin se conecta con la API de OpenAI para generar preguntas adaptativas.

Ya que Moodle almacena datos del *plugin* en su base de datos, es recomendable, aunque no imprescindible, nociones de lenguaje SQL, sobre todo referente a cómo interpretar esquemas y como realizar consultas y modificaciones.

Por último, si te has planteado colaborar en el desarrollo del *plugin*, es recomendable tener conocimientos de sistema de control de versiones, concretamente de Git, ya que el repositorio se encuentra alojado en GitHub.

2.3 Configuración de un entorno de desarrollo

A continuación, te mostramos cómo preparar un entorno para trabajar con el *plugin*.

2.3.1 Paso 1: Configura un servidor local

1. Instala XAMPP, WampServer o el servidor que prefieras.
2. Configura el servidor para que sea compatible con Moodle:
 - Activa extensiones necesarias de PHP como `curl`, `xml`, `intl`, `zip`, y `gd`.
 - Configura un acceso a una base de datos, por ejemplo MariaDB.

2.3.2 Paso 2: Instala Moodle

1. Descarga Moodle desde su [sitio oficial](#).
2. Sigue las instrucciones de instalación, asegurándote de crear una base de datos y un usuario con permisos adecuados.
3. Accede al administrador de Moodle para verificar que la instalación ha concluido con éxito.

2.3.3 Paso 3: Instala Composer

1. Una vez instalado, navega al directorio del plugin y ejecuta:

```
composer install
```

Esto descargará todas las dependencias definidas en el archivo `composer.json`.

2.3.4 Paso 4: Configura tu editor o IDE

1. Usa un editor de código moderno como Visual Studio Code o PhpStorm.
2. Instala extensiones útiles:
 - Formateadores y analizadores de código PHP.
 - *Plugins* específicos para Moodle, si están disponibles.

3 Estructura del *plugin*

DALEK está compuesto por una estructura organizada en carpetas y archivos que cumplen funciones específicas dentro del entorno Moodle. A continuación, se describe cada componente para entender su propósito y funcionalidad.

3.1 Descripción general

La estructura del *plugin* sigue el estándar de desarrollo de Moodle, lo que permite su integración con la plataforma. A continuación, se presenta el árbol de directorios y archivos principales del *plugin*, junto con una breve descripción de cada uno:

<code>backup/</code>	<code># Gestión de copias de seguridad y restauración.</code>
<code>classes/</code>	<code># Clases PHP, incluyendo eventos y lógica principal.</code>
<code>db/</code>	<code># Configuración de base de datos y permisos.</code>
<code>lang/</code>	<code># Archivos de idioma para la internacionalización.</code>
<code>vendor/</code>	<code># Dependencias externas gestionadas con Composer.</code>
<code>composer.json</code>	<code># Configuración de las dependencias del plugin.</code>
<code>lib.php</code>	<code># Funciones generales y API interna.</code>
<code>mod_form.php</code>	<code># Formulario de configuración del plugin.</code>
<code>settings.php</code>	<code># Configuración general accesible desde Moodle.</code>
<code>version.php</code>	<code># Información sobre la versión del plugin.</code>
<code>view.php</code>	<code># Archivo principal para mostrar la interfaz del plugin.</code>

3.1.1 backup/

Este directorio contiene los scripts necesarios para gestionar las copias de seguridad y restauraciones de Moodle relacionadas con el *plugin*.

- `backup_cuestionariollm_activity_task.class.php`: Define la tarea de copia de seguridad para las actividades generadas por el *plugin*.
- `restore_cuestionariollm_activity_task.class.php`: Configura cómo se restauran las actividades del *plugin* desde una copia de seguridad.
- `*_stepslib.php`: Contienen las instrucciones paso a paso para realizar las copias de seguridad y restauraciones.

3.1.2 classes/

En esta carpeta se encuentran las clases principales del *plugin*, organizadas según las convenciones de Moodle. Un subdirectorio típico dentro de esta carpeta es **event/**.

- **event/**: Contiene clases de eventos que Moodle utiliza para rastrear las interacciones del usuario con el *plugin*.
 - **course_module_instance_list_viewed.php**: Evento que registra cuándo un usuario visualiza la lista de actividades del *plugin*.
 - **course_module_viewed.php**: Evento que registra cuándo un usuario accede a una actividad específica.

Si deseas añadir nuevas funcionalidades al *plugin*, como eventos personalizados, este es el lugar donde debes incluir las clases correspondientes.

3.1.3 db/

La carpeta **db/** contiene archivos relacionados con la configuración de la base de datos y los permisos.

- **access.php**: Define las capacidades o permisos que los usuarios pueden tener al interactuar con el plugin. Por ejemplo, permisos para crear, editar o eliminar actividades.
- **install.xml**: Archivo XML que describe la estructura de las tablas utilizadas por el plugin en la base de datos de Moodle. Si necesitas añadir nuevas tablas o campos, este archivo debe ser actualizado.

3.1.4 lang/

Esta carpeta almacena los archivos de idioma necesarios para internacionalizar el **plugin**.

- **en/cuestionariollm.php**: Contiene las cadenas de texto en inglés para el *plugin*. Si deseas traducir el plugin a otros idiomas, debes crear carpetas como **es/** (español) o **fr/** (francés), copiando y adaptando este archivo.

3.1.5 vendor/

Esta carpeta contiene las bibliotecas externas gestionadas con Composer, necesarias para el funcionamiento del *plugin*. Algunas de las dependencias incluidas son:

- **guzzlehttp**: Biblioteca para realizar solicitudes HTTP hacia la API de OpenAI.
- **smalot/pdfparser**: Biblioteca para procesar archivos PDF y extraer texto.

- **psr7**: Implementación de estándares PSR para el manejo de solicitudes y respuestas HTTP.

Nota: No modifiques manualmente los archivos de esta carpeta. Si necesitas añadir o actualizar dependencias, utiliza Composer y el archivo `composer.json`.

3.1.6 Archivos principales

A continuación, se explican los archivos más importantes del *plugin*, que se encuentran en el nivel raíz:

composer.json: Contiene la configuración de las dependencias externas del *plugin*. Si necesitas añadir una nueva biblioteca, este archivo debe ser editado y actualizado con Composer. **lib.php**: Este archivo centraliza funciones genéricas del *plugin*. Aquí se pueden definir hooks que extiendan o personalicen el comportamiento de Moodle. **mod_form.php**: Archivo que define el formulario de configuración del *plugin* cuando se crea o edita una actividad en Moodle. Aquí puedes personalizar las opciones que el usuario ve al configurar la actividad. **settings.php**: Archivo utilizado para definir configuraciones generales del plugin, accesibles desde el panel de administración de Moodle. Por ejemplo, este archivo puede incluir la configuración de la API Key de OpenAI. **version.php**: Contiene la información sobre la versión del *plugin*, como el número de versión, la compatibilidad con Moodle y las dependencias requeridas. **view.php**: Es el archivo principal que renderiza la interfaz del plugin cuando los usuarios acceden a una actividad.

pix/

Esta carpeta almacena los recursos gráficos del *plugin*, como iconos. Por ejemplo:

- **monologo.svg**: Icono utilizado para representar el plugin en la interfaz de Moodle.

3.2 Buenas prácticas al trabajar con la estructura

- **No modificar directamente las dependencias de vendor/**: Usa Composer para gestionar estas bibliotecas.
- **Usar nombres descriptivos**: Los nombres de archivos y carpetas deben reflejar claramente su función.
- **Mantener la compatibilidad con Moodle**: Sigue las convenciones de desarrollo de Moodle para garantizar que el *plugin* funcione correctamente con futuras actualizaciones.

4 Instalación del *plugin*

La instalación del *plugin* en Moodle requiere seguir unos pasos sencillos pero cruciales para asegurar su funcionamiento correcto. A continuación, se describe detalladamente el proceso.

4.1 Descarga e instalación

4.1.1 Paso 1: Obtener los archivos del *plugin*

1. Descarga la versión más reciente del *plugin* desde el [repositorio oficial](#).
 - Asegúrate de que la versión del *plugin* sea compatible con la versión de Moodle que utilizas. DALEK requiere Moodle 4.1 o superior.
2. Comprueba que el archivo descargado esté en formato comprimido `.zip` o `.tar.gz`.

4.1.2 Paso 2: Copiar el *plugin* al directorio adecuado

1. Descomprime el archivo descargado en tu equipo.
2. Verifica que la carpeta raíz del *plugin* tenga un nombre distinto al resto de *plugins* instalados (por defecto es `dalek`). Si el nombre coincide con otro *plugin* que tenas instalado, cámbialo para que coincida.
3. Accede al directorio donde está instalado Moodle en tu servidor. Por ejemplo, si Moodle está instalado en `/var/www/moodle/`, utiliza esa ruta.
4. Copia la carpeta descomprimida del *plugin* (`dalek`) dentro del directorio `mod` de Moodle (por defecto en `/var/www/moodle/mod/`).

4.1.3 Paso 3: Comprobar los permisos de los archivos

1. Asegúrate de que el usuario del servidor web (normalmente `www-data` en sistemas GNU/Linux) tenga permisos de lectura y escritura sobre los archivos y carpetas del *plugin*.

2. Ejecuta el siguiente comando en la terminal para ajustar los permisos si fuera necesario:

```
sudo chown -R www-data:www-data /var/www/moodle/mod/dalek  
sudo chmod -R 755 /var/www/moodle/mod/dalek
```

4.2 Activación del *plugin* en Moodle

4.2.1 Paso 1: Acceder al panel de administración

1. Inicia sesión en Moodle como administrador.
2. Ve a la página de administración localizada en *Administración del sitio > Notificaciones*

4.2.2 Paso 2: Comprobar la instalación automática

1. Moodle detectará automáticamente el nuevo *plugin* en el directorio `mod` y mostrará una notificación indicando que hay un *plugin* listo para instalar.
2. Revisa los detalles de la instalación y confirma para continuar.

4.2.3 Paso 3: Revisión de requisitos

1. Moodle verificará que el servidor cumpla con los requisitos mínimos para el *plugin*:
 - Versión de Moodle: 4.1 o superior.
 - Versión de PHP: 8.1 o superior.
 - Extensiones de PHP necesarias, como curl y json.
2. Si algún requisito no se cumple, Moodle mostrará un error. Resuelve los problemas indicados antes de proceder.

4.2.4 Paso 4: Completar la instalación

1. Si todo está correcto, Moodle instalará automáticamente las tablas de base de datos necesarias para el *plugin*, utilizando el archivo `install.xml` incluido en la carpeta `db`.
2. Una vez finalizado, aparecerá un mensaje de confirmación indicando que el *plugin* se instaló correctamente.

4.3 Configuración inicial del plugin

4.3.1 Paso 1: Acceder a la configuración

1. Tras la instalación, accede a la configuración del plugin a través de *Administración del sitio > Plugins > Actividades > DALEK*

4.3.2 Paso 2: Configurar la API Key de OpenAI

DALEK requiere una API Key válida para conectarse al servicio de OpenAI y generar preguntas basadas en los textos aportados por los estudiantes.

1. Introduce tu API Key en el campo correspondiente. Si no tienes una API Key:
 - Regístrate en el [portal de OpenAI](#).
 - Genera una clave desde la sección de configuración de la API.
2. Guarda los cambios.

4.4 Comprobaciones finales

4.4.1 Paso 1: Verificar la funcionalidad

1. Crea un curso de prueba en Moodle y añade una actividad utilizando el *plugin* DALEK.
2. Sube un archivo PDF con contenido para generar un cuestionario.
3. Comprueba que el cuestionario se genera correctamente y que las preguntas son coherentes.

4.4.2 Paso 2: Habilitar la depuración (opcional)

1. Si encuentras errores, habilita el modo de depuración en Moodle para obtener más detalles en *Administración del sitio > Desarrollo > Depuración*

4.4.3 Paso 3: Probar diferentes configuraciones

Experimenta con distintas configuraciones de OpenAI y ajustes del *plugin* para asegurarte de que cumple con las necesidades de tu asignatura o curso.

4.5 Actualización del *plugin*

4.5.1 Paso 1: Obtener la nueva versión

Descarga la última versión del plugin desde el [repositorio oficial](#)

4.5.2 Paso 2: Reemplazar los archivos

1. Realiza una copia de seguridad (nunca viene mal) de la carpeta del *plugin* a reemplazar:

```
cp -r /var/www/moodle/mod/dalek /var/www/moodle/mod/dalek.old
```

2. Sustituye los archivos antiguos con los de la nueva versión.

4.5.3 Paso 3: Completar la actualización

1. Accede a *Administración del sitio > Notificaciones*
2. Moodle detectará automáticamente la actualización y aplicará los cambios necesarios en la base de datos.

4.5.4 Paso 4: Verificar compatibilidad

1. Comprueba que la nueva versión es totalmente funcional realizando pruebas en un curso.

4.6 Notas adicionales

- **Backup previo:** Aunque lo hemos explicitado, siempre **siempre**, antes de cualquier actualización o modificación importante, realiza una copia de seguridad completa de Moodle, incluyendo la base de datos.
- **Soporte técnico:** Si encuentras problemas durante la instalación o configuración, consulta la documentación oficial de Moodle o abre un *issue* en el repositorio del *plugin*.

Con estos pasos, el plugin DALEK estará completamente operativo en la instalación de Moodle. Si necesitas ayuda adicional, consulta la sección de soporte en esta guía.

5 Uso del plugin

En este capítulo explicamos cómo utilizar DALEK desde la perspectiva de un usuario que desee aprovechar al máximo sus funcionalidades para crear cuestionarios adaptativos basados en documentos cargados.

5.1 Introducción al flujo básico del *plugin*

El flujo de trabajo del *plugin* se basa en los siguientes pasos principales:

1. **Carga del documento base:** El profesor carga un documento (en formato PDF) que será utilizado como base para la generación de preguntas.
2. **Procesamiento del contenido:** El plugin analiza el contenido del documento y lo envía a un LLM a través de la API de OpenAI.
3. **Generación de preguntas:** El LLM devuelve un conjunto de preguntas adaptadas al contenido del documento.
4. **Configuración del cuestionario:** El profesor ajusta las opciones del cuestionario, como el número de preguntas y el formato de las mismas.
5. **Publicación y uso:** El cuestionario se publica en el curso de Moodle para que los estudiantes puedan resolverlo.

5.2 Añadir una nueva actividad de cuestionario adaptativo

5.2.1 Paso 1: Acceso a la configuración del curso

1. Entra en Moodle como profesor con permisos de edición.
2. Navega al curso donde desees añadir el cuestionario.
3. Activa el modo de edición haciendo clic en el botón correspondiente.

5.2.2 Paso 2: Añadir una actividad

1. Haz clic en *Añadir una actividad o un recurso* dentro del tema o sección deseada.
2. En el cuadro emergente, selecciona *Cuestionario LLM* y pulsa Añadir.

5.2.3 Paso 3: Configuración inicial

Se abrirá un formulario con las siguientes opciones principales:

- **Nombre del cuestionario:** Introduce un título descriptivo para la actividad (por ejemplo, “Cuestionario sobre Redes Neuronales Convolucionales”).
- **Descripción:** Proporciona una breve explicación del cuestionario para los estudiantes.
- **Archivo base:** Sube un archivo PDF que será procesado por el *plugin*. El archivo debe ser claro y estructurado, ya que el modelo de lenguaje utiliza este texto para generar las preguntas.
- **Número de preguntas:** Define cuántas preguntas deseas incluir en el cuestionario.
- **Idioma:** Selecciona el idioma del cuestionario. Por defecto, estará en español.

5.2.4 Paso 4: Guardar y procesar

1. Una vez completado el formulario, pulsa *Guardar y regresar* al curso o *Guardar y mostrar*.

En este punto el *plugin* procesará el archivo y generará automáticamente las preguntas.

5.3 Configuración avanzada de las preguntas

Después de la generación inicial, el profesor puede ajustar las preguntas generadas. Para ello:

1. Entra en la actividad recién creada.
2. Haz clic en el botón Editar preguntas.
3. Aparecerá una lista con las preguntas generadas. Cada pregunta incluye:
 - El **texto** de la pregunta.
 - **Opciones de respuesta:** En caso de preguntas de opción múltiple.
 - **Respuesta correcta:** Resaltada para facilitar la revisión.
4. Se dispone, en caso de que sea necesario, de las siguientes opciones de edición:
 - **Modificar el texto de la pregunta:** Pulsa el icono de edición junto a la pregunta para ajustarla.
 - **Añadir nuevas preguntas:** Haz clic en el botón Añadir pregunta manualmente para incluir preguntas no generadas por el *plugin*.
 - **Eliminar preguntas:** Marca las preguntas no deseadas y selecciona la opción de eliminar.

5.4 Publicación del cuestionario

Una vez revisadas y ajustadas las preguntas:

1. Dirígete a la página principal de la actividad.
2. Pulsa el botón Hacer visible si la actividad está oculta.
3. Comparte las instrucciones con los estudiantes en el foro del curso o mediante un mensaje directo.

5.5 Resolución del cuestionario por parte de los estudiantes

Desde la perspectiva del estudiante:

1. Al acceder al curso, verán la actividad con el nombre definido.
2. Al hacer clic en la actividad, se mostrará el cuestionario con las preguntas configuradas.
3. Una vez completado el cuestionario, los resultados se guardarán automáticamente en el sistema.

6 Desarrollo y personalización

Este capítulo ofrece una guía detallada para entender cómo podemos personalizar o extender el *plugin* de acuerdo a las necesidades de desarrollo.

6.1 Añadiendo funcionalidades

Veamos cómo añadir nuevas funcionalidades a través de varios ejemplos:

6.1.1 Añadir soporte para otro proveedor de LLM

1. Crear una clase para el nuevo proveedor:
 - Ubica la nueva clase en el directorio `classes/`
 - Define una interfaz común para interactuar con el proveedor. Por ejemplo:

```
namespace cuestionariollm\provider;

class CustomLLMProvider {
    public function __construct($apikey) {
        $this->apikey = $apikey;
    }

    public function generate_question($text) {
        // Implementar la lógica para conectarse al nuevo proveedor.
    }
}
```

2. Actualizar la lógica de generación de preguntas:
 - En `lib.php`, ajusta la función principal que genera las preguntas para permitir seleccionar entre proveedores.
3. Modificar la configuración:
 - Edita `settings.php` para añadir una opción que permita seleccionar el proveedor deseado desde la interfaz de Moodle.

6.1.2 Añadir nuevos eventos

1. Crea un archivo en `classes/event/` siguiendo el formato de los existentes. Por ejemplo, `course_module_deleted.php`:

```
namespace cuestionariollm\event;

defined('MOODLE_INTERNAL') || die();

class course_module_deleted extends \core\event\base {
    protected function init() {
        $this->data['crud'] = 'd';
        $this->data['level'] = self::LEVEL_TEACHING;
    }

    public static function get_name() {
        return get_string('eventcoursemoduledeleted', 'cuestionariollm');
    }

    public function get_description() {
        return "El módulo con id {$this->contextinstanceid} ha sido eliminado.";
    }
}
```

2. Registra el evento en Moodle para que quede disponible en el sistema.

6.2 Personalización de las preguntas generadas

La personalización de las preguntas generadas es la característica más interesantes del *plugin*, ya que permite adaptar las evaluaciones a las necesidades específicas de los cursos, liberando al docente de parte de la carga que ello implica.

6.2.1 Modificación de parámetros enviados a OpenAI

1. Localiza la función responsable:
 - En `lib.php`, busca la función que realiza la llamada a la API de OpenAI. Por ejemplo: “‘php \$response = \$client->post('https://api.openai.com/v1/completions', ['headers' => ['Authorization' => "Bearer \$apikey"], 'json' => ['model' => 'text-davinci-003', 'prompt' => \$prompt, 'max_tokens' => 200, 'temperature' => 0.7,]]);

2. Ajusta los parámetros:
 - Cambia valores como `temperature` para controlar la creatividad de las respuestas, o `max_tokens` para limitar la longitud.
3. Realiza varias pruebas con diferentes textos de entrada para verificar que los ajustes producen los resultados esperados.

6.2.2 Control del formato de las preguntas

1. Editar la plantilla de generación:
 - En el mismo archivo, modifica el contenido del *prompt* para ajustar el estilo de las preguntas generadas. Por ejemplo:

```
$prompt = "Genera una pregunta de opción múltiple basada en el siguiente texto: $text"
```

2. Añadir opciones avanzadas:
 - Por ejemplo, permitir que el usuario elija el tipo de pregunta (verdadero/falso, opción múltiple, ensayo) desde la configuración del módulo.

6.2.3 Modificación del esquema de base de datos

En algunos casos, es posible que desees extender la funcionalidad del *plugin* mediante nuevas tablas o campos en la base de datos. Para ello:

1. **Actualiza `install.xml`:** Añade una nueva tabla o un nuevo campo siguiendo el formato XML. Ejemplo de una nueva tabla:

```
<TABLE NAME="cuestionariollm_extra_data">
  <FIELDS>
    <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" SEQUENCE="true"/>
    <FIELD NAME="moduleid" TYPE="int" LENGTH="10" NOTNULL="true"/>
    <FIELD NAME="extra_info" TYPE="text" NOTNULL="false"/>
  </FIELDS>
  <KEYS>
    <KEY NAME="primary" TYPE="primary" FIELDS="id"/>
  </KEYS>
</TABLE>
```

2. **Crea funciones de acceso en `lib.php`:** Por ejemplo, para insertar datos:


```
function insert_extra_data($moduleid, $extra_info) {  
    global $DB;  
    $record = new stdClass();  
    $record->moduleid = $moduleid;  
    $record->extra_info = $extra_info;  
    $DB->insert_record('cuestionariollm_extra_data', $record);  
}
```

7 Solución de problemas

En este capítulo detallaremos cómo identificar y resolver los problemas más comunes que pueden surgir al instalar, configurar o usar el plugin DALEK en Moodle. Se incluyen en ellos los pasos para habilitar la depuración, interpretar mensajes de error y solucionar conflictos con dependencias u otros plugins.

7.1 Errores comunes y sus soluciones

7.1.1 Error durante la instalación del plugin

Al instalar el plugin desde el panel de administración de Moodle, aparece un mensaje de error indicando que faltan dependencias o que la versión de Moodle no es compatible.

- Causa/s posible:
 - Moodle no está actualizado a la versión mínima requerida (4.1).
 - No se han instalado correctamente las dependencias con Composer.
- Solución:
 1. Asegúrate de que Moodle está actualizado. Si es necesario, descarga la última versión.
 2. En la raíz del plugin, ejecuta:

```
composer install
```

Esto instalará las dependencias necesarias.

3. Reintenta la instalación desde el panel de administración.

7.1.2 El cuestionario no se genera correctamente

El plugin crea una actividad, pero no aparecen preguntas generadas automáticamente.

- Causa/s posible:
 - La API Key de OpenAI no está configurada o no es válida.

- El archivo cargado (PDF o texto) tiene un formato incompatible.
- Solución:
 1. Verifica que la API Key está configurada correctamente en `settings.php`. Prueba la conexión con:


```
$client = new \OpenAI\Client('tu_api_key');
```
 2. Asegúrate de que el archivo cargado tiene texto legible (no imágenes escaneadas). Puedes usar herramientas como `pdftotext` para verificar.
 3. Revisa los permisos del servidor para subir archivos. Moodle requiere que los archivos cargados cumplan con los límites de tamaño configurados en `php.ini` (`upload_max_filesize` y `post_max_size`).

7.1.3 Problemas de rendimiento en el servidor

Moodle se ralentiza notablemente cuando el *plugin* está en uso.

- Causa/s posible:
 - Las solicitudes a la API de OpenAI no están optimizadas o el servidor de Moodle tiene recursos limitados.
- Solución:
 1. Revisa los tiempos de respuesta de la API de OpenAI desde el archivo de registro.
 2. Si el problema persiste, habilita el almacenamiento en caché de las respuestas del modelo. Esto puede hacerse mediante una modificación en `lib.php` para guardar temporalmente las respuestas generadas.
 3. Considera optimizar la configuración de PHP en el servidor (por ejemplo, aumentando `max_execution_time` y `memory_limit`).

7.1.4 Conflictos con otros plugins

Algunas funciones del *plugin* no funcionan correctamente o Moodle muestra errores relacionados con otros *plugins*.

- Causa posible:
 - Otro plugin está sobrescribiendo eventos o funciones usadas por DALEK.
- Solución:
 1. Desactiva temporalmente los otros *plugins* sospechosos desde el panel de administración.

2. Revisa los eventos registrados en `classes/event/`. Si es necesario, modifica las clases para evitar conflictos de nombres.

7.2 Habilitar la depuración en Moodle

Cuando algo no funciona como se espera, habilitar la depuración en Moodle es una muy buena forma para ayudar a identificar la causa exacta del problema.

Los pasos para habilitar la depuración son los siguientes:

1. Accede a la configuración de depuración:
 - Ve a *Administración del sitio > Desarrollo > Depurador*.
2. Selecciona el nivel de depuración más alto: **DESARROLLADOR**.
3. Marca la casilla para mostrar mensajes de depuración directamente en la interfaz.
4. Guarda los cambios.
5. Reproduce el error y analiza los mensajes que aparecen en pantalla.

7.3 Uso de los registros (*logs*)

Moodle registra eventos y errores que pueden ser útiles para resolver problemas. Sus ubicaciones son las siguientes:

- Los registros del servidor web (Apache o Nginx) suelen estar en:
 - `/var/log/apache2/error.log` (Apache).
 - `/var/log/nginx/error.log` (Nginx).
- Moodle también tiene sus propios registros accesibles desde:
 - *Administración del sitio > Informes > Registros*.

¿Qué tendríamos que buscar:

- Errores específicos relacionados con DALEK, como:

`Exception: OpenAI API request failed with status 401 (Unauthorized)`
- Detalles de la actividad del plugin, como fallos al cargar archivos o errores en la generación de preguntas.

7.4 Solución de conflictos de dependencias

El *plugin* utiliza varias bibliotecas externas gestionadas con Composer. Si hay conflictos de versiones, podrían llegar a aparecer errores.

Pasos para resolver conflictos:

1. Reinstalar dependencias:

- Elimina el directorio `vendor/`:

```
rm -rf vendor/
```

2. Reinstala las dependencias:

```
composer install
```

3. Verifica la versión de las dependencias:

- Comprueba el archivo `composer.json` para asegurarte de que las versiones especificadas son compatibles.

4. Actualiza Composer y dependencias:

```
composer self-update # Actualiza Composer  
composer update      # Actualiza las dependencias
```

7.5 Preguntas frecuentes

7.5.1 ¿Por qué no puedo subir archivos grandes al plugin?

1. Verifica los límites en `php.ini`:

```
upload_max_filesize = 20M  
post_max_size = 20M
```

2. Reinicia el servidor después de realizar cambios.

7.5.2 ¿Qué hacer si la API de OpenAI no responde?

1. Comprueba el estado del servicio en su página de estado oficial.
2. Asegúrate de que el servidor tiene acceso a Internet.

7.5.3 ¿Cómo depurar errores relacionados con las preguntas generadas?

1. Activa los *logs* detallados en la API de OpenAI configurando el modo
2. de prueba (`test_mode`) en el *plugin*.
- 3.

8 Licencias y contribuciones

El plugin DALEK: Digital Analysis and Learning Enhancement Kit y su documentación están distribuidos bajo licencias que aseguran la libertad de uso, modificación y redistribución, respetando las normativas de software libre y acceso abierto.

A continuación, se detallan las licencias aplicables y las formas de contribuir al proyecto.

8.1 Licencia del código fuente: GPLv3

El código fuente del plugin está publicado bajo la Licencia Pública General de GNU versión 3 (GPLv3), que garantiza las siguientes libertades:

1. **Uso libre:** Puedes utilizar el *plugin* para cualquier propósito, sin restricciones de tipo comercial, académico o personal.
2. **Acceso al código fuente:** Puedes inspeccionar, estudiar y comprender cómo funciona el *plugin*. Esto fomenta la transparencia y la confianza en el software.
3. **Modificación:** Tienes el derecho de modificar el código fuente para adaptarlo a tus necesidades. Por ejemplo, puedes añadir nuevas funcionalidades, mejorar el rendimiento o personalizar su integración en Moodle.
4. **Distribución:** Puedes redistribuir el plugin original o tus versiones modificadas, siempre que respetes las mismas condiciones de la GPLv3, es decir:
 - Proporcionar acceso al código fuente.
 - Incluir una copia de la licencia GPLv3.
 - Asegurarte de que las modificaciones también estén licenciadas bajo GPLv3.

8.1.1 Obligaciones de la GPLv3

Si decides redistribuir el plugin o una versión modificada, debes:

- Incluir el aviso de copyright original.
- Acompañar el código fuente con una copia de la licencia GPLv3.
- Garantizar que las modificaciones sean públicas y accesibles bajo los mismos términos.

Para más información sobre la GPLv3, visita la web oficial de la Free Software Foundation](<https://www.gnu.org/licenses/gpl-3.0.html>).

8.2 Licencia de la documentación: CC-BY-SA

La documentación del plugin, incluyendo este manual, está licenciada bajo Creative Commons Attribution-ShareAlike 4.0 International (CC-BY-SA). Esto significa que:

- **Atribución (BY):** Puedes compartir (copiar y redistribuir) y adaptar (modificar o transformar) este documento para cualquier propósito, incluso comercial, siempre que reconozcas la autoría original. Esto implica:
 - Mencionar a los autores originales (por ejemplo, “Basado en el manual de DALEK”).
 - Incluir un enlace a la licencia.
 - Indicar si realizaste cambios en el documento.
- **Compartir igual (SA):** Si modificas o adaptas este documento, debes distribuir tu versión bajo la misma licencia CC-BY-SA. Esto asegura que cualquier mejora o variación sea accesible para otros usuarios en condiciones similares.
- **Sin restricciones adicionales:** No puedes imponer términos legales o tecnológicos que limiten los derechos concedidos por esta licencia.

Para más información sobre esta licencia, puedes visitar la web de la [licencia CC-BY-SA](#).

8.3 Contribuciones al proyecto

El desarrollo de DALEK está abierto a la colaboración de cualquier persona interesada en mejorar el plugin o aportar ideas. A continuación, se explica cómo puedes contribuir.

1. **Reportar problemas:** Si encuentras errores, problemas de compatibilidad o sugerencias de mejora, puedes abrir un *issue* en el [repositorio oficial en GitHub](#) del proyecto. Detalla claramente:
 - El problema encontrado.
 - Pasos para reproducirlo.
 - Mensajes de error (si los hay).
 - Tu entorno de trabajo (versión de Moodle, PHP, etc.).
2. **Proponer mejoras:** Si tienes ideas para nuevas funcionalidades o mejoras, abre un *issue* con una descripción detallada de tu propuesta.
3. **Aportar código:** Puedes realizar un *fork* del repositorio, trabajar en tus cambios y enviar un *pull request* (PR). Asegúrate de seguir las directrices de estilo y contribuir con código documentado y probado. Cada PR debe incluir:
 - Una descripción clara de los cambios realizados.
 - Explicación de cómo probar los cambios.
 - Referencia a cualquier issue relacionado.

4. **Traducir el *plugin*:** Si deseas colaborar con traducciones, puedes enviar archivos de idioma para otros idiomas además del español e inglés.
5. **Crear o mejorar la documentación:** Puedes aportar guías adicionales, ejemplos prácticos o tutoriales para mejorar la experiencia de otros usuarios y desarrolladores.

8.4 Reconocimientos

Agradecemos a todos los contribuyentes actuales y futuros por su esfuerzo y dedicación al proyecto. DALEK es una herramienta en constante evolución gracias a la colaboración abierta de la comunidad.

9 Referencias y recursos

En este capítulo se recogen las principales referencias y recursos que pueden ser de utilidad para el desarrollo, personalización y uso del *plugin*.

El objetivo es proporcionar información práctica y accesible para que los desarrolladores puedan resolver dudas, aprender nuevas funcionalidades o ampliar sus conocimientos sobre Moodle, PHP y las tecnologías empleadas en el *plugin*.

9.1 Moodle Developer Docs

Moodle es un sistema modular que ofrece múltiples APIs para interactuar con sus componentes principales. Es imprescindible conocer estas API para desarrollar plugins que sigan las mejores prácticas y mantengan la compatibilidad con futuras versiones de Moodle.

- **Documentación oficial**

<https://moodledev.io/>

Aquí se encuentra toda la documentación técnica de Moodle, incluyendo guías para desarrolladores, ejemplos de código y explicaciones detalladas de las APIs. Es el punto de partida para cualquier desarrollador interesado en trabajar con Moodle.

- **Foro de desarrolladores de Moodle**

<https://moodle.org/mod/forum/view.php?id=55>

Espacio para interactuar con la comunidad de desarrolladores de Moodle, resolver dudas y compartir ideas.

9.2 Repositorio del plugin

El repositorio oficial del plugin DALEK contiene el código fuente actualizado, así como instrucciones detalladas sobre la instalación, configuración y uso del plugin.

- **GitHub**

<https://github.com/KNODIS-Research-Group/moodle-dalek> En este repositorio también se gestionan los issues, pull requests y contribuciones al proyecto.

9.3 Documentación de OpenAI

El plugin utiliza la API de OpenAI para generar contenido adaptativo. La comprensión y el correcto uso de esta API son fundamentales para personalizar o extender las funcionalidades del plugin.

- **Guía de inicio rápido de OpenAI**
<https://platform.openai.com/docs/quickstart>
Proporciona ejemplos prácticos para comenzar a utilizar la API.
- **Referencia completa de la API**
<https://platform.openai.com/docs/api-reference>
Incluye información sobre los endpoints disponibles, parámetros y respuestas de la API.

9.4 Recursos sobre PHP y Composer

El desarrollo del plugin requiere bastante conocimiento de PHP, así como del uso de Composer para la gestión de dependencias.

- **Documentación oficial de PHP**
<https://www.php.net/docs.php>
Contiene guías completas, ejemplos y documentación de las funciones del lenguaje.
- **Documentación de Composer**
<https://getcomposer.org/doc/>
Explica cómo instalar y usar Composer para gestionar dependencias en proyectos PHP.

9.5 Recursos para el manejo de dependencias

El plugin utiliza varias dependencias externas, como `guzzlehttp` para realizar solicitudes HTTP y `smalot/pdfparser` para procesar archivos PDF. Aquí se incluyen los enlaces a sus respectivas documentaciones:

- **Guzzle (HTTP client)**
<https://docs.guzzlephp.org/>
Información sobre cómo realizar peticiones HTTP eficientes y manejar respuestas.
- **Smalot PDFParser**
<https://github.com/smalot/pdfparser>
Repositorio oficial con ejemplos y documentación para trabajar con archivos PDF.

9.6 Recursos adicionales

- **Guías de buenas prácticas en desarrollo PHP**
<https://phptherightway.com/>
Una recopilación de buenas prácticas y recomendaciones para desarrolladores PHP.
- **Control de versiones con Git**
 - <https://git-scm.com/doc>
Documentación oficial de Git para el control de versiones.
 - <https://learngitbranching.js.org/>
Tutorial interactivo para aprender a trabajar con ramas en Git.

9.7 Recursos para pruebas

La calidad del plugin se garantiza a través de pruebas exhaustivas. Moodle soporta PHPUnit para pruebas unitarias.

- **PHPUnit**
<https://phpunit.de/documentation.html>
Guía oficial para configurar y utilizar PHPUnit.
- **Pruebas específicas de Moodle**
<https://moodledev.io/docs/automation/phpunit>
Explicación de cómo configurar y ejecutar pruebas unitarias dentro de Moodle.