

Revisión y validación contenidos del Gestor de Contenidos vigencias pasadas

Objetivo

Validar la posibilidad de modificar y actualizar contenidos de un paquete SCORM exportado desde Adobe Captivate y posteriormente cargarlo nuevamente en Moodle.

Hallazgos Principales

1. Validación del paquete SCORM

Se identificó que el archivo inicialmente cargado en Moodle no era el SCORM real, sino un archivo ZIP contenedor que incluía otro ZIP.

Se verificó que el SCORM válido contenía:

- imsmanifest.xml
- index_scorm.html
- metadata.xml
- Carpetas assets, dr, vr, ar, entre otras.

Esto confirmó que el paquete cumplía con la estructura estándar SCORM 1.2.

2. Configuración de Moodle

Se aumentaron los límites de PHP para permitir la carga de paquetes de aproximadamente 160 MB.

Se modificaron parámetros como:

- upload_max_filesize
- post_max_size
- memory_limit

Posteriormente se reinició Apache para aplicar los cambios.

3. Duplicación del SCORM para pruebas

Antes de realizar modificaciones se creó una copia del SCORM dentro de Moodle.

Esto permitió:

- Mantener el contenido original intacto.
- Realizar pruebas sin afectar el curso funcional.

- Validar cambios de forma segura.

4. Análisis interno del paquete SCORM

Se descubrió que Adobe Captivate no almacena todas las imágenes como archivos PNG independientes.

En su lugar:

- Muchas imágenes se encuentran embebidas dentro de archivos JSON.
- Los archivos se encuentran principalmente en la carpeta:

dr

Por ejemplo:

img1.json

img2.json

img3.json

Cada archivo JSON contiene imágenes codificadas en Base64.

5. Extracción de imágenes

Se desarrolló un procedimiento para:

- Identificar imágenes utilizadas por cada diapositiva.
- Extraer imágenes PNG desde los archivos JSON.
- Generar una carpeta de trabajo con las imágenes exportadas.

Esto permitió visualizar y localizar fácilmente imágenes utilizadas dentro del contenido SCORM.

6. Identificación de imágenes y diapositivas

Se determinó que existe un archivo índice denominado:

imgmd.json

Este archivo relaciona:

Imagen → JSON que la contiene

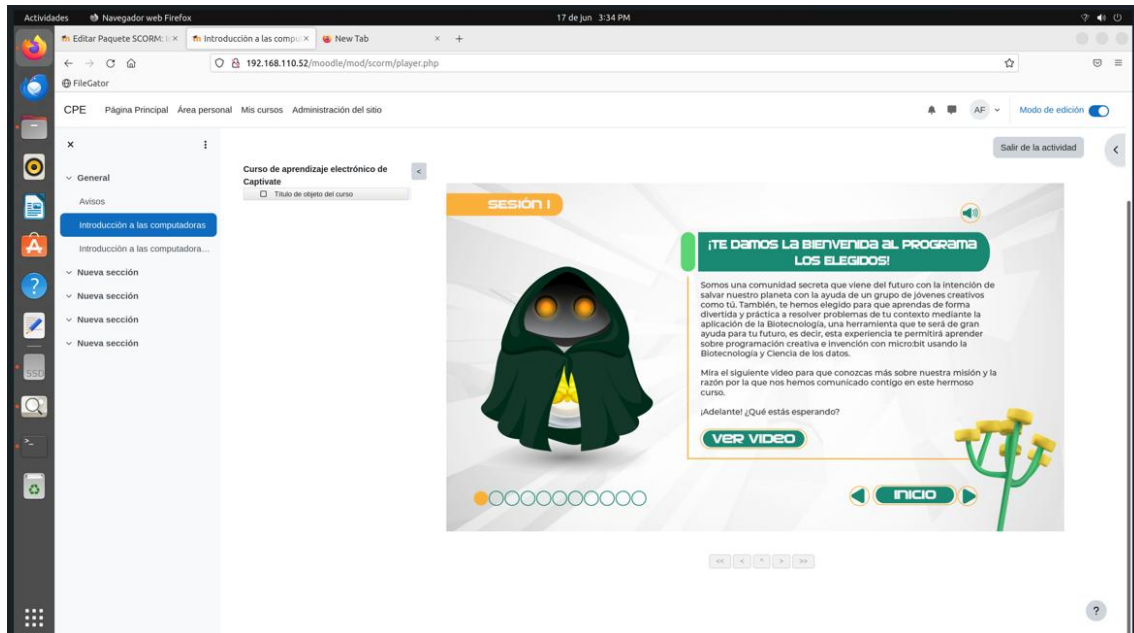
Ejemplo:

dr/8157_507_550.png → img51.json

Lo anterior permitió identificar exactamente qué archivo JSON debía modificarse para reemplazar una imagen específica.

7. Reemplazo de imágenes

Antes



Después

