

# KDiap

Un sistema de creación de paquetes SCORM enfocados a la  
resolución guiada de problemas.

Manual de uso

Miguel Samplón Chalmeta  
msamplon@unizar.es

20 de junio de 2009

## 1. ¿Qué es KDiap?

A fecha de hoy la gestión de la docencia con base no presencial o bien el apoyo de la docencia presencial mediante sistemas no presenciales se hace a través de un LMS (Learning Management System) que entre otras cuestiones se encarga de autenticar al usuario (generalmente mediante el uso de claves), gestionar los contenidos y su secuenciación temporal, evaluar y gestionar calificaciones y establecer canales de comunicación entre los usuarios. La interacción con el usuario se realiza a través de una interfaz Web. Ejemplos de LMS son Moodle y WebCT entre otros.

Cada LMS tiene su propia manera de organizar y administrar la información que gestiona, lo que conduce a que contenidos docentes desarrollados para un LMS no sean directamente interpretables por otro. Un LMS suele incorporar herramientas de importación/exportación de contenidos desde y hacia otras plataformas, si bien esto constituye una solución parcial y no completamente satisfactoria dado que no es raro que se produzcan pérdidas o desajustes en el proceso.

El estándar SCORM (Sharable Content Object Reference Model) nació con el propósito de homogeneizar la forma que en se construyen contenidos docentes. Toda la información necesaria se encapsula en un *paquete SCORM* que es reconocido y gestionado de igual forma por cualquier LMS que obedezca este estándar. Contempla al menos tres aspectos diferentes, aunque relacionados:

- La forma en que se van a secuenciar las diferentes partes del contenido docente.
- La forma en el que el contenido docente va a intercambiar información (nivel de progreso, calificaciones...) con el LMS.
- La forma en que se van a organizar los diferentes ficheros que componen el paquete SCORM.

SCORM se apoya un un conjunto amplio de tecnologías y/o estándares subyacentes: HTML, XML, DOM, Javascript... por lo que su utilización no es directa y requiere una cierta curva de aprendizaje. Por otra parte existe una complejidad inherente al hecho del que el estándar SCORM debe contemplar un abanico muy amplio de formas de situaciones docentes. En consecuencia la creación de contenidos SCORM por parte de un usuario final va a requerir el uso de herramientas informáticas (editores, entornos de desarrollo...) que simplifiquen la tarea y oculten total o parcialmente la necesidad de conocer los estándares subyacentes antes mencionados. Si bien ya han aparecido algunas de estas herramientas (Reload, Trident) todavía no han alcanzado un grado de madurez que las haga utilizables por cualquier usuario.

KDiap pretende establecer sistema sencillo y asequible para un usuario con una formación en informática a nivel de usuario de crear paquetes SCORM dirigidos a un conjunto *muy restringido* de situaciones. En concreto pretende ser una herramienta para poder crear contenidos conducentes a la

resolución guiada de un problema dentro del ámbito científico/técnico. El problema se va desarrollando paso a paso y con interacción con el usuario a través de resultados numéricos parciales. El resultado final será un paquete SCORM que encapsule ese problema.

## 2. Un ejemplo sencillo

En las imágenes que siguen se muestra un ejemplo sencillo en torno a un problema de polarización de un transistor desarrollado en 3 pasos (o diapositivas) más el paso final.

**Transistor NPN en activa**  
**Paso 1/3**

Se va a plantear un problema de polarización de transistores  
Durante el proceso se solicitará que se introduzcan algunos datos numéricos  
Los datos deberán ser validados mediante el boton rotulado como "HECHO"

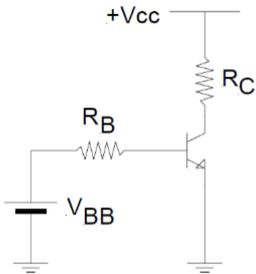
En caso de error se solicitará que se revisen los calculos y se introduzca un nuevo valor  
En caso de que la respuesta sea adecuada, se activará el botón "Continuar" que permitirá avanzar al siguiente paso.

Miguel Sampión Chalmeta  
[msampion@unizar.es](mailto:msampion@unizar.es)  
Universidad de Zaragoza

Figura 1: Ejemplo de polarización de un transistor. Figura 1

**Transistor NPN en activa**  
**Paso 2/3**

En el transistor de la figura se sabe que las fuentes de alimentación tienen un valor tal que el transistor se encuentra en modo activo



Qué valor tiene la tensión base- emisor

Tensión base - emisor [V]

Tensión base - emisor [V] está mal calculado  
-----  
Formulario completado incorrectamente.  
Revise los resultados y vuelva a intentarlo.

Miguel Sampión Chalmeta  
[msampion@unizar.es](mailto:msampion@unizar.es)  
Universidad de Zaragoza

Figura 2: Ejemplo de polarización de un transistor. Figura 2

## 3. El paquete SCORM KDiap

Un paquete SCORM se construye colocando en un directorio todos los ficheros que componen los contenidos docentes, típicamente páginas HTML junto con el resto de elementos que van em-

### Transistor NPN en activa

#### Paso 3/3

Acabamos de ver que la tensión  $V_{be}$  es de 0.7 V  
 Sabiendo que la corriente de colector es de 35 mA y que la beta del transistor es de 250. ¿Cuál es la corriente de base y la de emisor?

Corriente de base [mA]

Corriente de emisor [mA]

---

Formulario completado correctamente.  
 Pulse el botón "siguiente" para proseguir.

Miguel Sampión Chalmeta  
[msampion@unizar.es](mailto:msampion@unizar.es)  
 Universidad de Zaragoza

Figura 3: Ejemplo de polarización de un transistor. Figura 3

### Final del problema

La actividad se ha completado y evaluado  
 Puedes cerrar esta ventana para acabar

Miguel Sampión Chalmeta  
[msampion@unizar.es](mailto:msampion@unizar.es)  
 Universidad de Zaragoza

Figura 4: Ejemplo de polarización de un transistor. Figura 4

bebidos en esas páginas: imágenes, vídeos, scripts... Junto a ellos va un fichero XML denominado `imsmanifest.xml` que describe la estructura y secuenciación de contenidos junto con un conjunto de esquemas XML que le dan soporte. El conjunto de ficheros (mínimo) que va a constituir un paquete SCORM KDiap se listan a continuación:

```

adlcp_v1p3.xsd
adlnav_v1p3.xsd
adlseq_v1p3.xsd
datatypes.dtd
imscp_v1p1.xsd
imsmanifest.xml
imsss_v1p0.xsd
imsss_v1p0auxresource.xsd
imsss_v1p0control.xsd
imsss_v1p0delivery.xsd
imsss_v1p0limit.xsd
imsss_v1p0objective.xsd
imsss_v1p0random.xsd
imsss_v1p0rollup.xsd
imsss_v1p0sequelrule.xsd
imsss_v1p0util.xsd
ims_xml.xsd
KBarra.png
KDefinicionDiap.js
KGestorDiap.js
KHEstilo.css
KPaginaDiapBase.htm
ostyn2004sco.js
xml.xsd
XMLSchema.dtd

```

### 3.1. Los ficheros que no deben modificarse

La mayor parte de esto fichero deben formar parte del paquete pero no deben modificarse. Cabe mencionar aqui el fichero `ostyn2004sco.js`, que es un conjunto de funciones escritas por Claude Ostyn en javascript para facilitar la interacción con el LMS.

### 3.2. Los ficheros que pueden modificarse

#### 3.2.1. El fichero `KPaginaDiapBase.htm`

Este fichero corresponde a una página html que será el marco en el que se visualizará todo el contenido. Se puede escribir como se quiera si bien debe cumplir los siguientes requisitos.

- Debe incluir en la seccion `<HEAD>` las tres líneas siguientes

---

```
<SCRIPT TYPE="text/javascript" SRC="ostyn2004sco.js"></SCRIPT>
<SCRIPT TYPE="text/javascript" SRC="KGestorDiap.js"></SCRIPT>
<SCRIPT TYPE="text/javascript" SRC="KDefinicionDiap.js"></SCRIPT>
```

---

- Debe contener 5 bloques del tipo

---

```
<DIV ID="#####">
</DIV>
```

---

con los siguientes campos ID:

- `K_TITULO`: Dentro de este bloque se escribirán los diferentes títulos
  - `K_ENUNCIADO`: Dentro de este bloque se escribirá el enunciado del problema (o parte del problema)
  - `K_FORMULARIO`: Dentro de este bloque se mostrará el formulario para la entrada de datos
  - `K_RESULTADO`: Dentro de este bloque aparecerá el resultado de la evaluación del formulario
  - `K_NAVEGACION`: Dentro de estos bloques aparecerán los botones de navegación
- El elemento `<BODY>` debe tener el controlador del evento de carga de la página asociado a la función `_KAlCargarLaPagina()`

---

```
<BODY ONLOAD="_KAlCargarLaPagina()">
```

---

A continuación se transcribe una página básica (aunque no mínima) con todo lo anterior.

---

```

</HTML>
<HEAD>
<TITLE>KDiap. Página base</TITLE>
<LINK REL="stylesheet" TYPE="text/css" HREF="KHEstilo.css">
<SCRIPT TYPE="text/javascript" SRC="ostyn2004sco.js"></SCRIPT>
<SCRIPT TYPE="text/javascript" SRC="KGestorDiap.js"></SCRIPT>
<SCRIPT TYPE="text/javascript" SRC="KDefinicionDiap.js"></SCRIPT>
</HEAD>
<BODY CLASS="KBody" ONLOAD="_KAlCargarLaPagina()" >
<IMG SRC="KBarra.png" ALT="barra">
<!-- TITULO -->
<DIV ID="K_TITULO">
</DIV>
<IMG SRC="KBarra.png" ALT="barra">
<!-- ENUNCIADO -->
<DIV ID="K_ENUNCIADO">
</DIV>
<!-- FORMULARIO -->
<DIV ID="K_FORMULARIO">
</DIV>
<IMG SRC="KBarra.png" ALT="barra">
<!-- RESULTADO -->
<DIV ID="K_RESULTADO">
</DIV>
<IMG SRC="KBarra.png" ALT="barra">
<!-- NAVEGACION -->
<DIV ID="K_NAVEGACION">
</DIV>
<IMG SRC="KBarra.png" ALT="barra">
<DIV CLASS="KBloque" ID="PIE">
<P CLASS="Pie">Autor de la página</P>
<P CLASS="Pie">
<A HREF="mailto:correo@electronico.es">correo@electronico.es</A>
</P>
<P CLASS="Pie">Universidad de Zaragoza</P>
</DIV>
</BODY>
</HTML>

```

---

Para dar flexibilidad y riqueza al diseño se ha incluido una hoja de estilo `KHEstilo.css` y una imagen `png KBarra.png` correspondiente a una barra con propósito decorativo. Estos dos ficheros son opcionales, pero de usarlos deben estar incluidos en el paquete SCORM. A continuación se transcribe un posible contenido del fichero `KHEstilo.css`

---

```

BODY.KBody
width: 750px ;

/** ESTILO PARA LOS BLOQUES ESTRUCTURALES **/
DIV.KBloque

width: 100% ;
margin-top: 5px ;
margin-bottom: 0px ;

/** TIPO PARA LOS PIES DE PAGINA **/
P.Pie

font-family: Times, serif ;
font-size: 8pt ;
font-weight: normal ;
text-align: left ;
text-indent: 1% ;
margin-top: 0 ;
margin-bottom: 0 ;

```

---

### 3.3. Los ficheros que deben modificarse

Son dos, el fichero `KDefinicionDiap.js`, que establece el contenido en sí del problema y el fichero `imsmanifest.xml` que define la estructura del paquete SCORM. Dada la importancia de cada uno de ellos, los expondremos en una sección propia

## 4. El fichero `KDefinicionDiap.js`

Este fichero describe cada una de las diapositivas a través de llamadas a funciones javascript. Para evitar que el usuario tenga que tener una formación sólida en javascript se han generado un conjunto de funciones (definidas en el fichero `KGestorDiap.js`) que constituyen una API simple. Pasamos a revisarlas:

- **KNumeroTotalDiapositivas (Num)** Establece el número total de diapositivas (sin incluir la diapositiva final). Las diapositivas se numeran de la 1 hasta la numero total. La diapositiva final tiene el número 0
- **KRegistraFuncionValidacion (IdxDiapo, Funcion)** Asocia una función de validación a una diapositiva. La función de validación no debe tener parámetros y debe devolver un booleano (true para ok, false para fallo)

Por ejemplo:

---

```
function UnaFuncionDeValidacionDeDiapositiva()
...
return true ;
...
return false ;
```

---

La asociamos a la diapositiva 1 como

---

```
KRegistraFuncionValidacion(1,UnaFuncionDeValidacionDeDiapositiva)
```

---

- **KTextoTitulo (IdxDiapo, Texto, TipoFormato)**

Añade la línea de titulo Texto con formato TipoFormato a la zona de título de la diapositiva IdxDiapo. Se escribirá un texto Texto en un nodo P con una clase TipoFormato definida en la hoja de estilo

---

```
<P CLASS=TipoFormato>Texto</P>
```

---

En caso de omitir el TipoFormato se usará uno por defecto

- **KImagenTitulo (IdxDiapo, Fichero, Anchura)** Añade la imagen contenida en el fichero Fichero con anchura porcentual Anchura a la zona de título de la diapositiva IdxDiapo

Se pintará una imagen en un nodo P con una anchura establecida en tanto por ciento Anchura. Si se omite el parámetro anchura se omitira el atributo WIDTH (la imagen se mostrará a su tamaño original)

---

```
<P> <IMG SRC=Fichero WIDTH=Anchura%> </P>
```

---

- **KTextoEnunciado (IdxDiapo, Texto, TipoFormato)**

Añade la línea de texto Texto con formato TipoFormato a la zona de enunciado de la diapositiva IdxDiapo. Se escribirá un texto Texto en un nodo P con una clase TipoFormato definida en la hoja de estilo

---

```
<P CLASS=TipoFormato>Texto</P>
```

---

En caso de omitir el TipoFormato se usará uno por defecto Si se quiere hacer referencia al valor de campo recogido en un formulario con anterioridad, el nombre del campo debe incluirse en el texto enmarcado entre signos dobles de \$\$

---

```
KTextoEnunciado(5,"El valor del campo ResultadoParcial_1 ha sido $$ResultadoParcial_1$$")
```

---

- **KImagenEnunciado (IdxDiapo, Fichero, Anchura)**

Añade la imagen contenida en el fichero Fichero con anchura porcentual Anchura a la zona de enunciado de la diapositiva IdxDiapo. Se pintará una imagen en un nodo P con una anchura establecida en tanto por ciento Anchura. Si se omite el parámetro anchura se omitirá el atributo WIDTH (la imagen se mostrará a su tamaño original)

---

```
<P> <IMG SRC=Fichero WIDTH=Anchura%> </P>
```

---

- **KDefineCampoFormulario (IdxDiapo, Nombre, Texto, ValorInicial, TipoFormato)**

Añade un campo de formulario a la diapositiva IdxDiapo. Los campos siempre representan número flotantes. El identificador del campo es Nombre. Irá acompañado de un texto Texto escrito con un formato TipoFormato. Tendrá un valor inicial ValorInicial (0.0) si se omite el parámetro. En caso de omitir el TipoFormato se usará uno por defecto.

- **KValorCampo (Nombre)** Función para ser usada dentro de las funciones de validación de diapositiva. Obtiene el valor del campo de formulario con nombre (ID) Nombre

- **KTextoResultado (Texto, TipoFormato)**

Función para ser usada dentro de las funciones de validación de diapositiva. Escribirá una línea de texto (en la diapositiva que esté activa en ese momento (típicamente la diapositiva asociada a la función de validación) con formato TipoFormato. En caso de omitir el TipoFormato se usará uno por defecto

- **KAproximadamenteIgual (Valor, Resultado, Tolerancia)**

Testea si la diferencia relativa de un número Valor con respecto a un número Resultado es inferior a una tolerancia Tolerancia, expresada en En concreto testea si:

---

```
abs((100*Valor-Resultado)/Resultado) < Tolerancia
```

---

Como caso particular, si Resultado es 0.0 testea

---

```
abs(100*Valor) < Tolerancia
```

---

devuelve true o false

- **KEstableceNotaFinal (Nota)**

Establece la nota final con que se valorará el ejercicio. Este valor puede alterarse cuanto se quiera; la nota se hará definitiva (y se trasladará al LMS) cuando se llegue a la diapositiva final.

## 5. El fichero **imsmanifest.xml**

Este fichero debe tener el siguiente contenido:

---

```

<?xml version="1.0" standalone="no"?>

<manifest identifier="(Nota 1)" version="(Nota 2)"
xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3"
xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3"
xmlns:imsss="http://www.imsglobal.org/xsd/imsss"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
http://www.imsglobal.org/xsd/imsss imsss_v1p0.xsd
http://www.adlnet.org/xsd/adlcp_v1p3 adlcp_v1p3.xsd
http://www.adlnet.org/xsd/adlseq_v1p3 adlseq_v1p3.xsd
http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p3.xsd">

<metadata>
<schema>ADL SCORM</schema>
<schemaversion>2004 3rd Edition</schemaversion>
</metadata>

<organizations default="KDiapOrgBase">
<organization identifier="KDiapOrgBase">
<title>(Nota 3)</title>

<item identifier="K232323" identifierref="KDiap">
<title>(Nota 4)</title>
</item>
</organization>
</organizations>

<resources>
<resource identifier="KDiap" adlcp::scormType="sco" type="webcontent" href="KPaginaDiapBase.htm">
<file href="KPaginaDiapBase.htm" />
<file href="KGestorDiap.js" />
<file href="KDefinicionDiap.js" />
<file href="ostyn2004sco.js" />
<file href="KBarra.png" />
<file href="KHEstilo.css" />
<file href="Nota 5" />
</resource>
</resources>

</manifest>

```

---

- Nota 1

Colocar una cadena de texto que sirva de identificador único. Debe comenzar con una letra. No tiene por qué tener sentido. Por ejemplo podría ser:

- Nota 2

Cualquier cadena de texto que sirva como identificación de la versión

---

```

<manifest identifier="Aklsiosksjioe23nmk23j1" version="0.0"

```

---

- Nota 3

- Nota 4

El título (descriptivo) de la actividad.

- Nota 5

Deben añadirse tantas filas de este tipo como archivos se auxiliares (imágenes png, jpg, etc...) se incluyan en las diapositivas.

## 6. SCORM en el LMS Moodle

Una vez reunidos todos los fichero que van a componer el paquete SCORM KDiap, deben comprimirse en formato zip. El fichero zip resultante constituye el paquete.

Para cargar el paquete en Moodle, debe subirse a la plataforma, a través de la opción Archivos y despues debe agregarsea uno de los bloques en que este dividido el curso como Agregar Actividad ->SCORM

## 7. Nota final

El paquete KDiap se ha desarrollado dentro del Programa de Enseñanza Semipresencial de la Universidad de Zaragoza. PESUZ 2008.

## 8. Bibliografía y recursos

- SCORM 2004: Content Aggregation Model (CAM) book.
- SCORM 2004: Sequencing and Navigation (SN) book.
- SCORM 2004: Run Time Environment (RTE) book.

Constituyen (parte) del estandar. Disponibles en:

---

<http://www.adlnet.org/Pages/Default.aspx>

---

- In the Eye of the SCORM. An introduction to SCORM 2004 for Content Developers. Claude Ostyn.

Muy recomendable. En la página que se cita a continuación aparecen más recursos sobre SCORM, además del fichero `ostyn2004sco.js`

---

<http://www.adlnet.org/Pages/Default.aspx>

---

- JavaScript. La Guía Definitiva. David Flanagan. Ed. O'Reilly.
- Professional javascript for web developers. Nicholas C. Zakas. Wiley Publishing Inc. Guías de JavaScript y DOM (Document Object Model).
- XML in a Nutshell. Elliotte Rusty Harold, W. Scott Means. Ed. O'Reilly.
- XML Schema. Eric van der Vlist. Ed. O'Reilly Guías de XML
- Editor Reload. <http://www.reload.ac.uk/> Editor de paquetes SCORM gratuito.