



III Seminario de Innovación Docente en Automática

INTEGRACIÓN DE WEBLABS EN LMS

Ildefonso Ruano Ruano

Universidad de Jaén



ÍNDICE

1. Introducción
2. [Integración Lab-LMS. Modos](#)
3. [Ejemplo: Integración de Lab JS EJS](#)



1. Introducción

E-learning (aprendizaje electrónico o docencia virtual): Tipo de educación en la que se emplean medios electrónicos en el proceso formativo [79].

Mayor exponente del uso de las TICs en la educación: LMS (*Learning Management System*), herramienta esencial de educación universitaria. Entorno Web ofrece **recursos y servicios**:

Espacio Virtual personalizado

Herramientas para...

...comunicarse (foros, chats, videoconferencias...)

...crear contenidos (wikis, blogs, ...)

...recoger opiniones de usuarios (encuestas)

...evaluar

Gestión de usuarios

Recursos de aprendizaje en diferentes formatos (*pods*, pdfs, ficheros, vídeos,...)

Seguimiento del progreso de aprendizaje

Objetivo Principal

Integración avanzada de laboratorios docentes *online* en LMSs

Laboratorios Online
(VRL)

Sistemas de Gestión
de Aprendizaje (LMS)



INTEGRACIÓN
BENEFICIOS

ÍNDICE

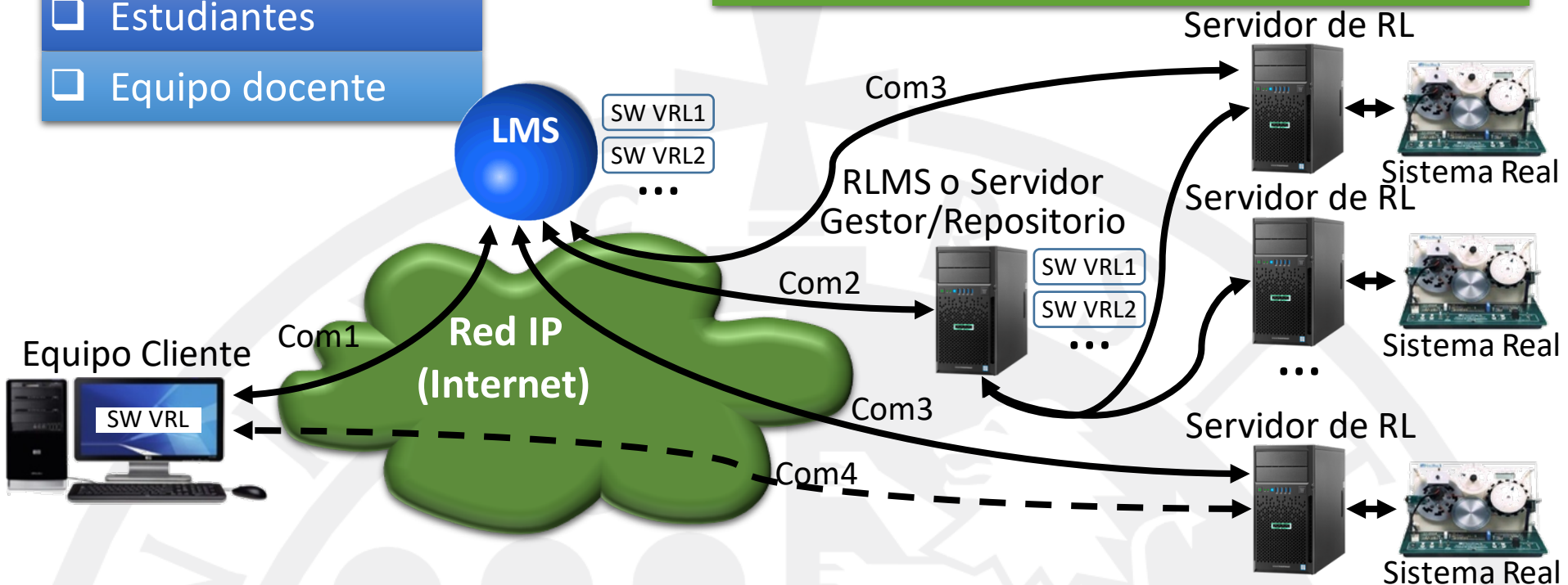
1. [Introducción](#)
2. Integración Lab-LMS. Modos
3. [Ejemplo: Integración de Lab JS EJS](#)



Roles en Integración Lab-LMS

- ☐ Creadores de VRL
- ☐ Estudiantes
- ☐ Equipo docente

Tecnologías de desarrollo y comunicación utilizadas



No perciben diferencia con otros contenidos del LMS

Relaciones con otros recursos docentes del LMS y Seguimiento

Prestaciones de Integración Lab-LMS

1. Identificación de alumnos

2. Ejecución en entorno conocido

3. Presentación con otros elementos del LMS

4. Establecimiento de secuencias de aprendizaje

5. Ejecución Personalizada/Accesible

6. Formación adaptada al alumno

7. Seguimiento del trabajo realizado en el VRL

8. Almacenamiento de resultados y calificaciones

Modos de Integración Lab-LMS y características

Integración	Modo	Ubicación	Comunicación con LMS	Recursos Asociados		Identificación	Portabilidad	Personalizar
				Pre-VRL	Post-VRL			
Nula	0	Ext	No	No	No	No	Sí	No
Mínima	1	Ext. (enlace)	No	No	No	No	No	No
Media	2	LMS	No	Posible	No	LMS	No	No
Avanzada	3	LMS	Estándar	Posible	Posible	LMS	Posible	Posible
	4	LMS	Propia	Posible	Posible	LMS	No	Posible
	5	Ext. y LMS	Estándar	Posible	Posible	VRL-LMS	Posible	Posible

Ubicación

Identificación

Portabilidad

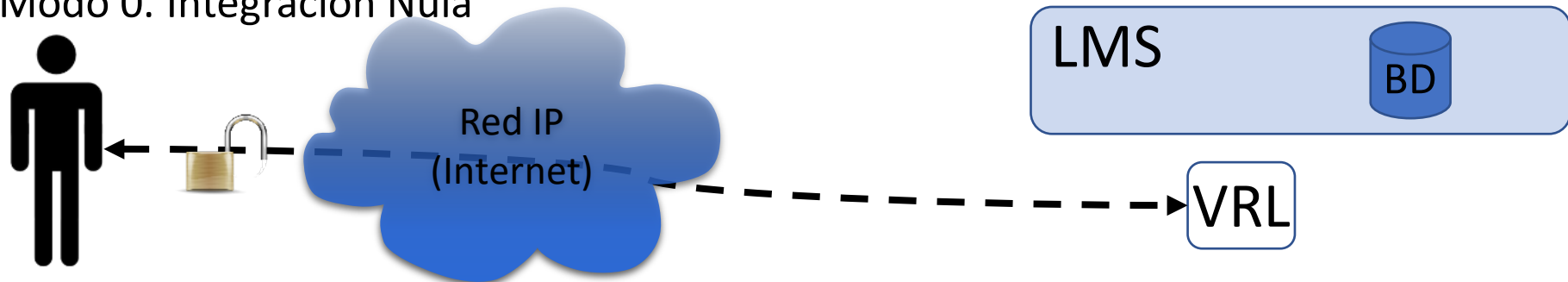
Secuencias de aprendizaje

Personalizar aprendizaje

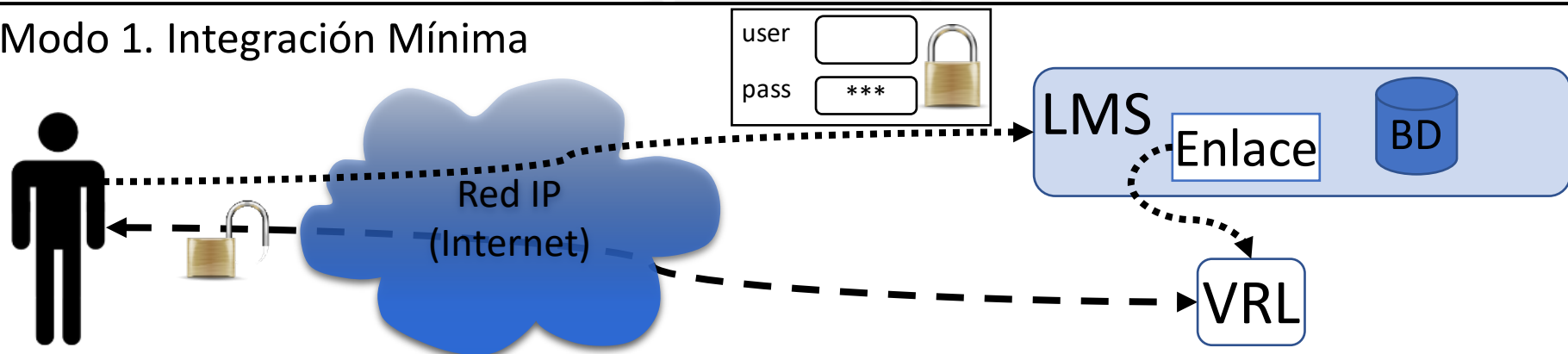
Comunicación

2. Integración Lab-LMS

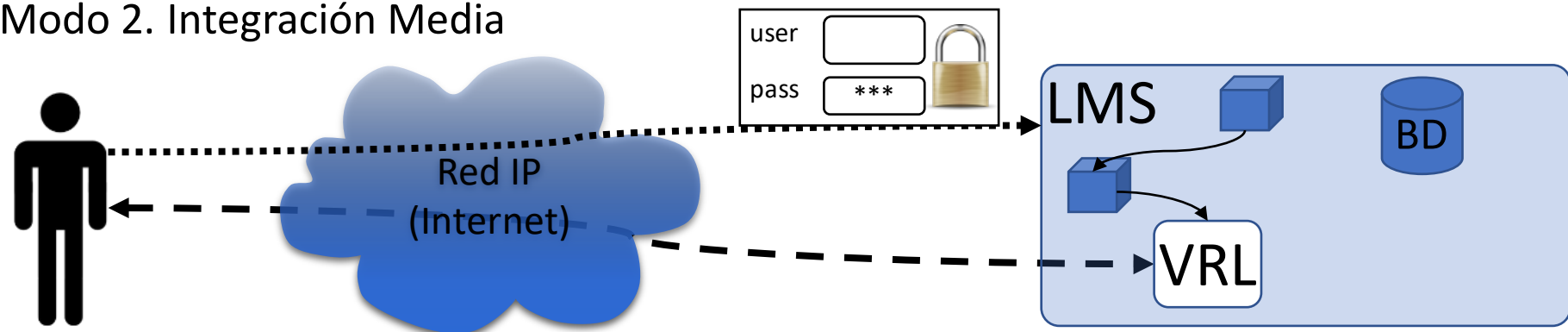
Modo 0. Integración Nula



Modo 1. Integración Mínima

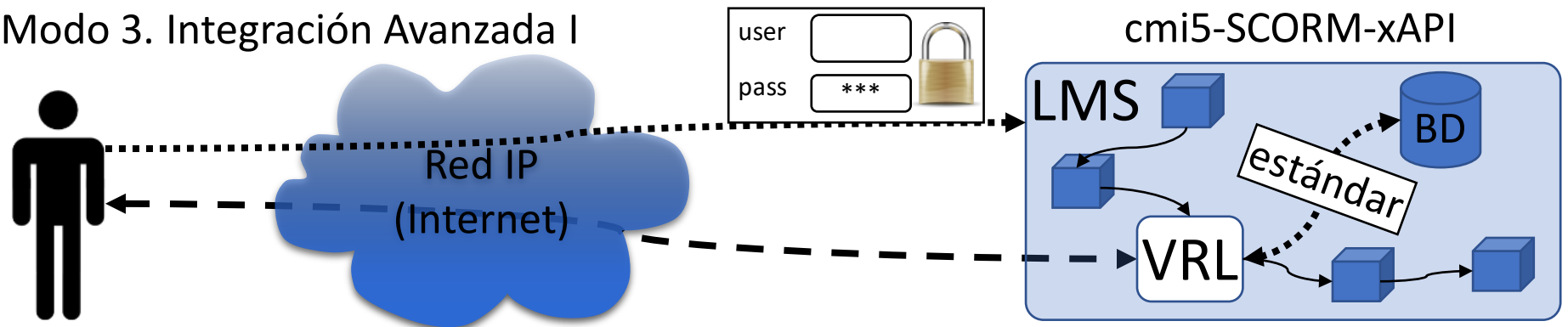


Modo 2. Integración Media

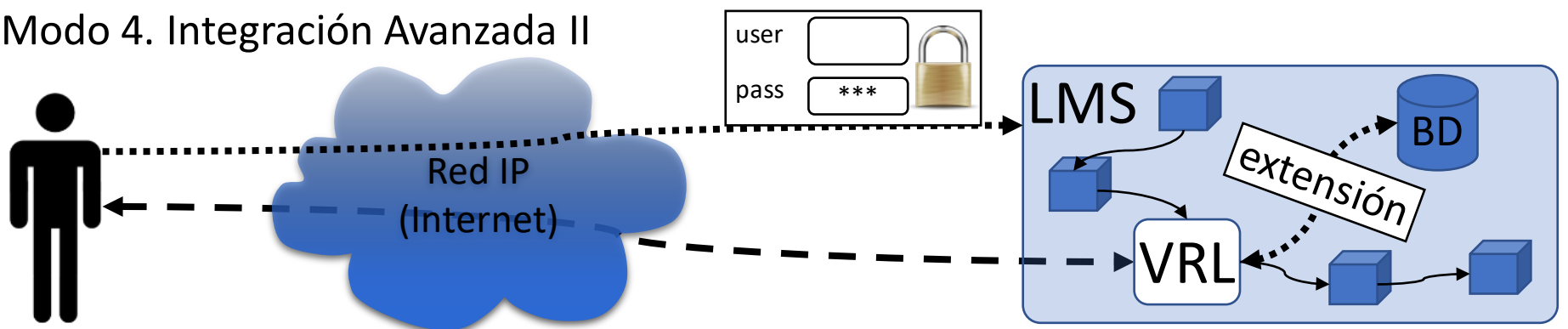


2. Integración Lab-LMS

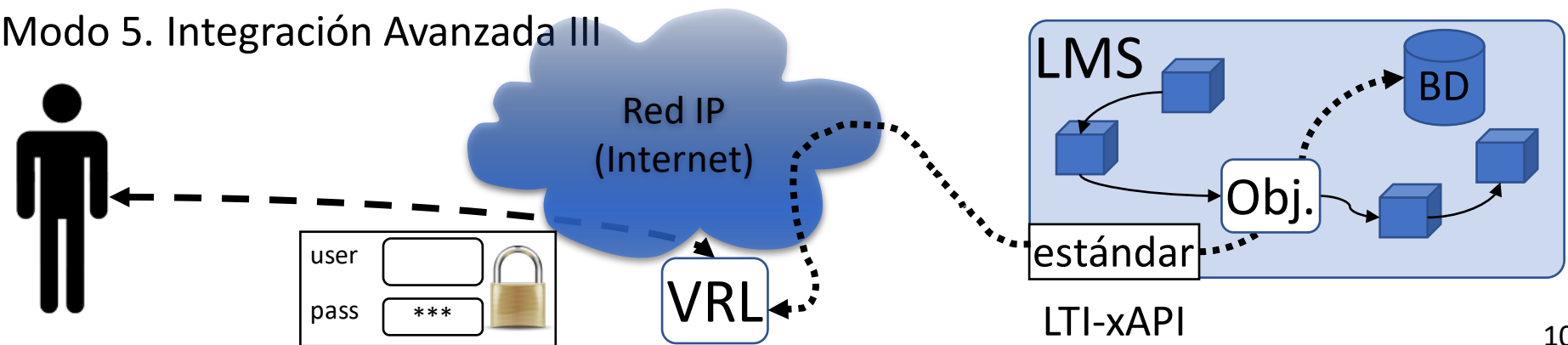
Modo 3. Integración Avanzada I



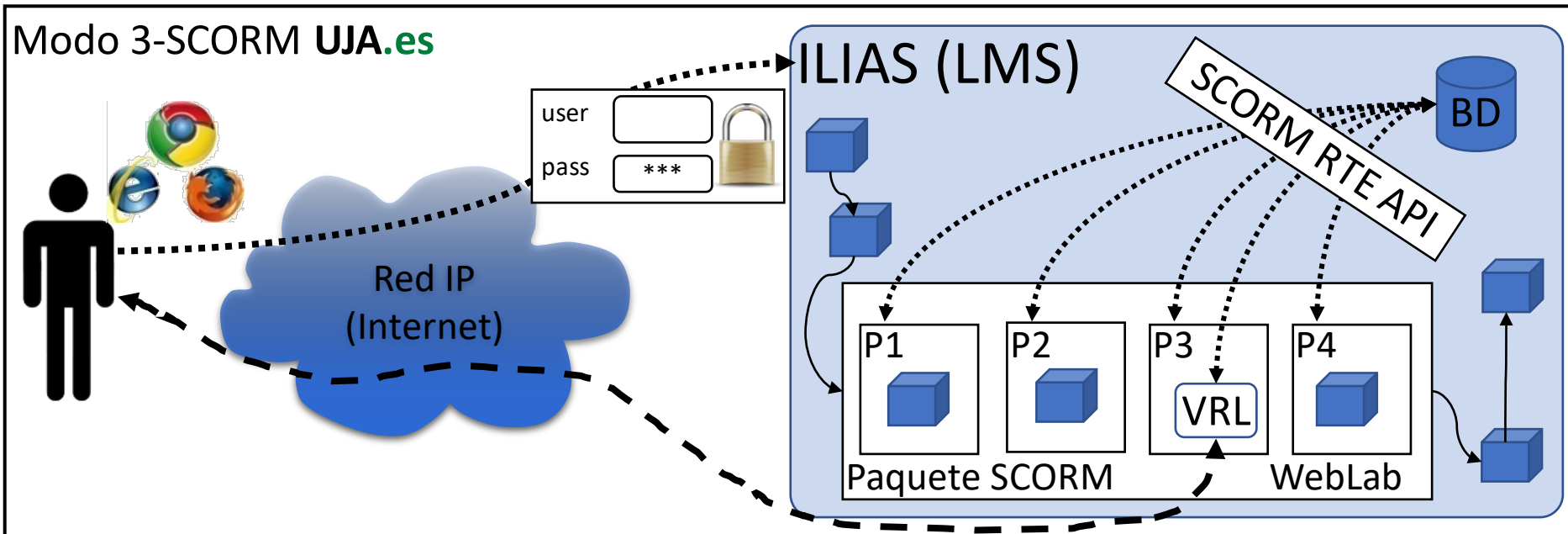
Modo 4. Integración Avanzada II



Modo 5. Integración Avanzada III



Propuesta de Integración Avanzada (Modo 3)



- ❑ Todas las características avanzadas.
- ❑ Basado en estándares SCORM (paquetes modelo).
- ❑ Recursos asociados dentro y fuera de SCORM.

SCORM (*Shared Content Object Reference Model*)

- ❑ Estándar de facto de contenido de *e-learning*.
- ❑ Proporciona marco en contenidos para definir su encapsulación, lanzamiento e intercambio de datos.

Sub-Especificaciones:

- ❑ **Modelo de Agregación de Contenido (CAM).** Especifica como describir y empaquetar contenido.
- ❑ **Secuenciación y Navegación (SN).** Especifica navegación entre partes de contenido (SCO o Asset).
- ❑ **Entorno de Tiempo de Ejecución (RTE).** Especifica como debe lanzarse y comunicarse con el LMS. Basado en ECMAScript (JavaScript).

ÍNDICE

1. [Introducción](#)
2. [Integración Lab-LMS. Modos](#)
3. **Ejemplo: Integración de Lab JS EJS**



Objetivo

Integrar simulación JS EJS en un LMS usando SCORM y herramientas de soporte desarrolladas:

Modelos SCORM de 1-4 páginas con librerías JS

Página Web de herramientas desarrolladas:

http://dv.ujaen.es/goto_docencia_fold_682679.html

Se explican en:

Ruano-Ruano, I.; Cano-Marchal, P.; Gámez-García, J.; Gómez-Ortega, J. “Advanced LMS Integration of SCORM Web Laboratories”. IEEE Access, vol.4 pp. 6352 - 6363

doi:10.1109/ACCESS.2016.2587805. [Web](#)

Proceso

1. Crear simulación JS en EJSS (ej. MassAndSpringComplete)
2. Exportar la simulación (paquete ZIP)
3. Descomprimir en una carpeta
4. Descomprimir modelo SCORM en la carpeta (ej. SCORM12-4p).
5. Modificar fichero imsmanifest.xml de SCORM
6. Modificar fichero de simulación: (ej.
MassAndSpringComplete_Simulation.xhtmll)
7. Comprimir todo en ZIP
8. Importar en LMS

Modificaciones de Fichero imsmanifest.xml

Sustituir nombre del fichero donde se desee meter la simulación

MassAndSpringComplete_Simulation.xhtml

```
</organizations>
<resources>
  <resource identifier="RES1" type="webcontent" adlcp:scormtype="sco" href="sco1.html">
    <file href="sco1.html"/>
    <file href="_ejs_library/APIWrapper.js"/>
    <file href="_ejs_library/RTE.js"/>
  </resource>
  <resource identifier="RES2" type="webcontent" adlcp:scormtype="sco" href="sco2.html">
    <file href="sco2.html"/>
    <file href="_ejs_library/APIWrapper.js"/>
    <file href="_ejs_library/RTE.js"/>
  </resource>
  <resource identifier="RES3" type="webcontent" adlcp:scormtype="sco" href="sco3.html">
    <file href="sco3.html"/>
    <file href="_ejs_library/APIWrapper.js"/>
    <file href="_ejs_library/RTE.js"/>
  </resource>
  <resource identifier="RES4" type="webcontent" adlcp:scormtype="sco" href="sco4.html">
    <file href="sco4.html"/>
    <file href="_ejs_library/APIWrapper.js"/>
    <file href="_ejs_library/RTE.js"/>
  </resource>
</resources>
```


Modificaciones de Fichero de Simulación xhtml

```
<script src="_ejs_library/scripts/APIWrapper.js"></script>
<script src="_ejs_library/scripts/RTE.js"></script>
```

Carga de funciones JS para
usar comunicaciones SCORM

```
<script type="text/javascript">
var miRTE = new RTE("1.2");
var terminated = false;
function unloadPage() {
    if (terminated != true) {
        miRTE.setExit("normal");
        miRTE.setSessionTime();
        miRTE.Terminate();
        terminated = true;
    }
}
</script>
```

Función *unloadPage()*

Asegurar cierre de comunicaciones

Inicializar comunicación y
llamadas a funciones de
RTE.js

```
<body onbeforeunload="unloadPage()">
```

```
var result = miRTE.initialize();
miRTE.setCompletionStatus("incomplete");
alert("Mi Nombre: " + miRTE.getLearnerName() + "\nMi Identificador en el LMS:" + miRTE.getLearnerId());
```

GRACIAS POR SU ATENCIÓN

