

Guía en Español del Run-Time Environment de SCORM 1.2 de la ADL

1.1 APPLICATION PROGRAM INTERFACE (API)

El API Adapter es el módulo que permite la comunicación entre el LMS (sistema de aprendizaje) y los SCO (contenidos intercambiables). Dicha comunicación está estandarizada.

Dicho de una forma sencilla el API simplemente es una librería de funciones a las que el SCO puede acceder. La forma en que el API adapter está programado no importa siempre que éste cumpla las especificaciones del estándar.

Cuando se trabaja con un SCO éste adquiere y manda información al LMS. Por tanto toda la comunicación entre el API adapter y el SCO la inicia el SCO.

Hay tres clases de funciones en el API:

-De estado de ejecución:

Las funciones de estado de ejecución son LMSInitialize("") y LMSFinish("")

-De estado de administración de errores:

Son LMSGetLastError(), LMSGetErrorString(errornumber) y LMSGetDiagnostic(parameter)

-Para transferir datos:

Son el resto de las funciones del API Adapter y mandan o reciben datos del LMS. Son LMSGetValue(data model element) , LMSSetValue(data model element, value) y LMSCommit("").

1.2 FUNCIONES DE ESTADO DE EJECUCIÓN

1.2.1 LMSInitialize():

Descripción: esta función indica al API Adapter que el SCO se va a comunicar con el LMS. Es obligatorio que el SCO llame primero a esta función antes que a ninguna otra del LMS.

Sintaxis: LMSInitialize(parámetro)

Parámetro: en esta versión sólo se permite meter el parámetro cadena vacía. Otros parámetros están reservados para futuras versiones.

Valor que devuelve la función: devuelve una cadena que representa un valor boolean

-True: indica que LMSInitialize() se ha iniciado con éxito.

-False: indica que LMSInitialize() no se ha iniciado con éxito. El SCO no ha sido bien identificado por el LMS y cualquier llamada adicional a una función del API no será procesada por el LMS.

1.2.2 LMSFinish():

Descripción: el SCO debe llamar a esta función cuando determine que ya no necesita comunicarse con el LMS más. Esta llamada asegura que:

El SCO puede asegurarse que cualquier dato usado por la función LMSSetValue() ha sido manejado por el LMS

El SCO ha finalizado su comunicación con el LMS

Sintaxis: LMSFinish(parámetro)

Parámetro: "" Una cadena vacía para este estándar. Otros valores distintos a "" han sido reservados para futuras versiones.

Valor devuelto: devuelve una cadena representando un valor Booleano

-True: indica que la función se ha ejecutado con éxito. Una vez devuelto este valor el SCO no puede llamar a otras funciones del API.

-False: indica que la función no se ha ejecutado con éxito. El LMS está en un estado desconocido y que cualquier llamada al LMS puede o no puede ser ejecutada.

1.3 FUNCIONES DE TRANSFERENCIA DE DATOS

1.3.1 LMSGetValue:

Descripción: esta función permite al SCO obtener información desde el LMS. Se usa para determinar:

- Valores para varias categorías (grupos) y elementos en el modelo de datos.
- La versión de modelo de datos soportada.
- Si una categoría o elemento específico es soportado o no.
- El número de elementos que en un momento dado hay en un array o una matriz.

Sintaxis: LMSGetValue(parámetro)

Parámetro: debe ser el nombre completo de la variable cuyo valor actualizado queremos obtener. Dicho valor es devuelto por la función en forma de string.

Según el tipo de dato:

- datamodel.group.element: devuelve el valor del elemento nombrado.
- datamodel._versión: la palabra clave _version se usa para determinar la versión del modelo de datos que soporta el LMS.
- datamodel.element._count: la palabra clave _count determina el número total de elementos actuales en un array.
- datamodel.element._children: la palabra clave _children determina todos los elementos en un grupo o categoría que sea compatible con el LMS.

Valor devuelto: todos los valores devueltos son cadenas.

- LMSGetValue(datamodel.group.element)

El valor devuelto es una cadena representando el valor actual del elemento o grupo pedido.

- LMSGetValue(datamodel._versión)

El valor devuelto es una cadena representando la versión del modelo de datos del LMS

- LMSGetValue(datamodel.group._children)

EL valor devuelto es una lista separada por comas de todos los nombres de los elementos en una categoría o un grupo específico que admite el LMS. Si no hay ningún elemento pero se admiten la función devuelve "". Si no se admiten elementos también se devuelve "". En ese caso puedo utilizar la función LMSGetLastError() para diferenciar si el elemento no es admitido.

- LMSGetValue(datamodel.group._count)

El valor devuelto es un integer que indica el número de objetos en una lista de elementos o en un array

- LMSGetValue(datamodel.group._count):

El valor devuelto es un integer que indica el número de objetos que actualmente hay en una lista o en un array.

1.3.2 LMSSetValue():

Descripción: Esta función permite al SCO enviar información al LMS. El adaptador API puede ser diseñado para remitir la información al LMS o a otro diseño relacionado con él.

Esta función se usa para poner los valores actuales de varias categorías o grupos y elementos en el modelo de datos.

Sintaxis: LMSSetValue(parámetro, valor)

Parámetro: El nombre del modelo de datos y su grupo se da como primer parámetro de la función. El argumento es case-sensitive (diferencia entre letras mayúsculas y minúsculas) y debe ser una cadena encerrada en comillas.

Exponemos algunas formas que este parámetro podría tomar:

- datamodel.element

Este es el nombre de la categoría o grupo definido en el modelo de datos. Un ejemplo es “cmi.comments”

- datamodel.group.element:

Este es el nombre de un elemento definido en el modelo de datos. Un ejemplo es “cmi.core.lesson_status”.

- Datamodel.group.n.element:

El valor del subelemento en el enésimo lugar de un elemento array (el primer valor es cero)

Valor: El valor del parámetro se da como segundo argumento de la función. En cada llamada a la función sólo se puede enviar un valor.

Valor devuelto: se devuelve una cadena representando un valor booleano.

-True: indica que la función se ha ejecutado con éxito

-False: indica que la función no se ha ejecutado con éxito.

1.3.3 LMSCommit()

Descripción: Si el adaptador API está cogiendo valores recibidos del SCO via la función LMSSetValue(), una llamada a LMSCommit() hace que cualquier valor no guardado en disco se almacene en disco, es decir, que sea persistente.

En algunas implementaciones, el adaptador API guarda los datos según los recibe y por tanto no los almacena en el caché del cliente. En éstas implementaciones no sería necesario utilizar esta función.

Sintaxis: LMSCommit(parámetro)

Parámetro: “”. Una cadena vacía es el parámetro para este estándar. Otros valores distintos están reservados para futuras extensiones.

Valor devuelto: Cadena representando un valor booleano.

-true: indica que la función se ha ejecutado con éxito.

-false: indica que la función no se ha ejecutado con éxito. El LMS se encontraría en un estado desconocido y cualquier otra llamada a una función del API podría o no podría ser procesada por el LMS.

1.4 FUNCIONES DE ESTADO DE GESTIÓN DE ERRORES

1.4.1 LMSGetLastError()

Descripción: El SCO debe tener una forma de saber si las funciones llamadas han sido ejecutadas correctamente y si no lo han sido, saber porqué han fallado. Esta función devuelve los códigos de error que resultan de la anterior llamada a una función del API. Cada vez que se llama a una función del API el código de error se borra y se reinicia pero si no se hace ninguna llamada al API el código de error permanece.

Sintaxis: LMSGetLastError()

Parámetro: ninguno

Valor devuelto: los valores devueltos son cadenas que se pueden convertir en números enteros que identifican errores según las siguientes categorías:

- 100: errores generales.
- 200: errores de sintaxis.
- 300: errores del LMS
- 400: errores del modelo de datos

Los siguientes códigos están disponibles para mensajes de error:

- 0: no hay error
- 101: excepción general.
- 201: error de argumento inválido
- 202: el elemento no puede tener hijos
- 203: el elemento no es un array y por tanto no se puede utilizar count.
- 301: No inicializado
- 401: error no implementado
- 402: el elemento es una palabra clave y no se le puede insertar ningún valor.

- 403: el elemento es de sólo lectura.
- 404: el elemento es de sólo escritura.
- 405: tipo de dato incorrecto

1.4.2 LMSGetErrorString:

Descripción: esta función obtiene una descripción textual del error representado por el código de error.

Sintaxis: LMSGetErrorString(errornumber)

Parámetro: un número entero representando un código de error.

Valor devuelto: una cadena representando la descripción del error.

1.4.3 LMSGetDiagnostic(parámetro)

Descripción: esta función activa las descripciones contenidas en el LMS para solucionar el error. De esta forma se obtiene mayor documentación para solucionar el error.

Sintaxis: LMSGetDiagnostic(parámetro)

Parámetro: el parámetro puede tomar una de estas dos formas

- Un número entero representando un código de error. Esto pide más información sobre el error consultado.
- "" – una cadena vacía - . Esto pide más información sobre el último error ocurrido.

Valor devuelto: el valor devuelto es una cadena que representa cualquier información sobre el error

1.5 COMUNICACIÓN DEL SCO Y EL LMS MEDIANTE EL API

La comunicación de un SCO con el LMS mediante el API adapter pasa por varios estados para una instancia determinada del SCO durante el tiempo de ejecución.

Los distintos estados del API adapter especifica una respuesta determinada del API adapter para un input dado. Durante cada uno de estos estados hay diferentes actividades por donde un SCO puede pasar.

Los estados en los que puede encontrarse el API son: no inicializado, inicializado y finalizado.

-No inicializado :

Describe el estado del SCO en el que este está entre el lanzamiento real y antes de que se ejecute LMSInitialize(""). Durante este estado el SCO tiene la responsabilidad de encontrar el API adapter suministrado por el LMS. Una vez que el API Adapter ha sido encontrado por el SCO, se permite que el SCO invoque las siguientes funciones del API

- LMSInitialize("")
- LMSGetLastError()
- LMSGetErrorString()
- LMSGetDiagnostic()

-Inicializado:

Describe el estado en que el SCO está situado entre la ejecución de LMSInitialize("") y LMSFinish("").

-Finalizado:

Describe el estado en que el SCO ha llamado a la función LMSFinish(""). Si el API adapter devuelve "false" al ser invocado LMSFinish("") entonces al SCO se le permite llamar a:

- LMSGetLastError()
- LMSGetErrorString()
- LMSGetDiagnostic()

Si el API adapter devuelve "false" no hay garantía de que éste responda correctamente a cualquier función.

1.6 USO DEL CÓDIGO DE ERROR DEL API

El SCO debe tener una forma de valorar si cualquier función de llamada al API se ha ejecutado con éxito y si no ha sido así, determinar cuál fue el error.

Con la función LMSGetLastError() averiguamos el número de error de la función anterior.

Los distintos valores que esta función puede devolver son:

Cód.	Descripción	Utilidad
"0"	No hay error	No se ha encontrado ningún error. La función se ha ejecutado perfectamente
"101"	Excepción general	Usado para indicar funciones generales.

“201”	Argumento inválido	Se usa cuando hay una llamada a una función del modelo de datos del SCORM Run Time Environment que no existe. Se usa cuando un argumento inválido pasa por el API
“202”	Los elementos no pueden tener hijos	Se usa cuando se llama a LMSGetValue() en cualquier modelo de datos que no soporte hijos.
“203”	El elemento no es un array y por tanto no se puede ejecutar “count”	Se usa cuando se llama a LMSGetValue() en cualquier categoría o elemento de un modelo de datos que no soporte count.
“301”	No inicializado	Se usa cuando se llama a cualquier función del API antes de ejecutar LMSInitialize(“”)
“401”	Error no implementado	Se usa cuando se hace una llamada al modelo de datos que no es soportada por el LMS o si se está usando otro modelo de datos fuera del SCORM Run-Time Environment Data Model.
“402”	El elemento es una palabra clave y por tanto no se le puede asignar dicho valor	Se usa cuando una llamada a LMSSetValue() implica utilizar una palabra clave.
“403”	El elemento es de sólo lectura	Se usa cuando una llamada a LMSSetValue() implica utilizar un elemento de sólo lectura.
“404”	El elemento es de sólo escritura	Se usa cuando una llamada a LMSSetValue() implica utilizar un elemento de solo escritura.
“405”	Tipo de dato incorrecto	Se usa cuando se hace un intento de asignar un valor a una variable que no corresponde con su tipo.

1.7 REGLAS GENERALES DEL API

La siguiente lista resume las reglas generales del API:

- En los nombres de las funciones hay diferencias entre mayúsculas y minúsculas (case-sensitive) y deben expresarse siempre tal y como son mostrados aquí.
- Idem con los parámetros y argumentos. Los parámetros siempre se expresan en minúsculas.

- Cada llamada a una función del API borra el código de error (a no ser que la función llamada se de administración de errores.

1.8 RESPONSABILIDADES DEL LMS

Las responsabilidades para usar en el LMS el adaptador API son las siguientes:

- El LMS debe lanzar el SCO en una ventana contenida en un marco principal donde esté el API adapter.
- El API adapter debe ser suministrado por el LMS.
- El acceso al API desde el SCO debe ser mediante Java-Script
- El adaptador API debe ser accesible via DOM como un objeto llamado “API”

1.9 ENCONTRAR EL API

El SCO tiene la responsabilidad de lanzar la llamada a LMSInitialize(“”) y LMSFinish(“”) al API. Para hacer esto el contenido debe ser capaz de localizar el API adapter que se presenta con el LMS.

Es responsabilidad del LMS suministrar un API adapter en la ventana que contenga la jerarquía DOM de forma que el SCO pueda buscar el marco principal de forma recursiva y/o abrir la ventana con la jerarquía DOM para encontrar el API.

Es responsabilidad del contenido encontrar y establecer comunicaciones con el adaptador API del LMS. La forma de hacerlo no está estandarizarla por SCORM.

1.10 MODELO DE DATOS

1.10.1 Introducción al modelo de datos

El propósito de establecer un modelo de datos común es asegurarse que la información sobre el SCO pueda ser seguida por diferentes LMS. Si, por ejemplo, se determina que seguir la puntuación de un alumno es un requerimiento del sistema, entonces es necesario establecer una vía común para el contenido para informar al LMS de las puntuaciones.. Si los SCO usan su propio sistema de puntuaciones, los sistemas de aprendizaje no sabrían como recibir, almacenar o procesar la información.

Hay un número de modelo de datos bajo desarrollo en varias comunidades y organizaciones estándar. Este borrador de las especificaciones del modelo de datos intenta agrupar funcionalmente la información que va a ser intercambiada entre el SCO y el LMS. Los ejemplos incluyen: interfaz con información del estudiante, preguntas y autoevaluación,

información del estado del sistema, valoración, etc.. En el lanzamiento de la actual versión de SCORM este borrador de las especificaciones del modelo está todavía bajo desarrollo.

1.10.2 Modelo de datos del entorno de ejecución en SCORM

El modelo de datos del entorno de ejecución en la versión actual de SCORM deriva directamente del modelo de datos de AICC CMI. Es posible que en un futuro se admitan más modelos de datos en SCORM.

Para identificar el modelo de datos en uso, todos los nombres de los elementos descritos en esta sección empiezan por “cmi”. Así señalamos que los elementos provienen del modelo de datos de CMI. Otros modelos de datos empezarán de otra forma cuando sean desarrollados.

1.10.3 Reglas Generales del entorno de ejecución del modelo de datos de SCORM

La siguiente lista resume las reglas generales:

- El primer símbolo en el nombre del elemento de datos identifica el modelo de datos. “cmi” indica modelo de datos CMI. Esto aumenta la funcionalidad del API permitiéndole manejar otros modelos de datos.
- Hay tres palabras reservadas. Son todas minúsculas y van precedidas de un carácter de subrayado.
- _versión: palabra clave usada para determinar la versión del modelo de datos que admite el LMS.
- _children: palabra clave usada para determinar qué elementos del modelo de datos son admitidos por el LMS.
- _count: palabra clave usada para determinar el número de elementos que hay en un momento dado en una lista.
- Todos los arrays empiezan a contar elementos por cero.
- Los nombres del modelo de datos diferencian entre mayúsculas y minúsculas (case-sensitive)
- El modelo de datos está implementado en un SCO. Un SCO no puede acceder los elementos de datos de otro SCO.

1.10.4 Elementos del modelo de datos

Los elementos del modelo de datos están divididos en 2 categorías: obligatorios y opcionales. Que elementos son obligatorios y cuales son opcionales viene especificado en la guía AICC CMI001 para compatibilidad entre 2 ordenadores.

Los elementos de datos obligatorios deben ser admitidos por el LMS. Además el LMS también puede proporcionar implementación para admitir algunos o todos los elementos del modelo de datos opcionales.

El uso de todos los elementos de datos por parte del SCO es opcional. Sólo es obligatorio que el SCO use las funciones del API LMSInitialize(“”) y LMSFinish(“”).

1.10.5 Gestión de listas

Para obtener o introducir un valor en una lista debemos utilizar el número índice. La única vez que un número índice se puede omitir es cuando hay solamente un elemento en la lista.

Podemos utilizar la palabra reservada `_count` para saber el número de elementos de una lista.

Si el SCO no conoce el número de registros sobre objetivos en un array, puede empezar la cuenta del estudiante actual con cero. Esto sobrescribiría cualquier información almacenada sobre objetivos que estuviera almacenada en primera posición. Que se sobrescriba o que se anexe es una decisión que hace el creador del SCO.

Se puede hacer llamadas a los elementos de una lista mediante la notación “.n”. Por ejemplo “cmi.objective.3.status”

1.10.6 Modelo de datos del entorno de ejecución desarrollado

cmi.core La información requerida debe ser soportada por todos los LMS. Hijos del cmi.core: student_id, student_name, lesson_location, credit, lesson_status, entry, score, total_time, lesson_mode, exit, session_time	
cmi.core._children	
Llamadas soportadas por el API: LMSGetValue()	Definición: La palabra clave <code>_children</code> se usa para determinar todos los elementos en la categoría principal que soporta el LMS. Si un elemento no tiene hijos (children) pero los soporta, se devuelve una cadena vacía. Si el elemento no es soportado por el LMS también se devuelve una cadena vacía pero podemos darnos cuenta haciendo un pedido sobre el último error en el cual se dirá que el elemento no es soportado por el LMS. Uso: Para determinar qué elementos de datos de cmi.core están admitidos por el LMS. Formato: El valor devuelto es una lista separada con comas del nombre de los elementos en la categoría principal que están soportados por el LMS Comportamiento del LMS: <ul style="list-style-type: none"> • Inicialización: El conjunto de hijos soportados para este grupo de set of
Obligatorio en LMS: Sí	
Tipo de dato: CMISString255	
Accesibilidad del SCO: Sólo lectura	

	<p>supported children for this group. Por tanto una llamada a LMSGetValue() devolvería la lista apropiada de hijos soportados.</p> <ul style="list-style-type: none"> • LMSGetValue(): LMS devuelve una lista separada por comas de los elementos admitidos. <ul style="list-style-type: none"> ○ Llamada de ejemplo al API: LMSGetValue("cmi.core._children") ○ Ejemplo de valores devueltos: "student_id,student_name,lesson_location,credit,lesson_status,entry, score,total_time, exit,session_time" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error de no implementación. Si el elemento cmi.core._children no es soportado por el LMS se devolvería una cadena vacía. NOTA: el cmi.core._children debe ser admitido por el LMS ya que el elemento es obligatorio. • LMSSetValue(): El LMS debería establecer un código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 402 – Valor establecido inválido: el elemento es una palabra clave. Si el elemento es admitido por el LMS (el elemento debe ser admitido ya que el elemento es obligatorio y una llamada a LMSSetValue sobre este elemento debe colocar el código de error en 402. ▪ 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. NOTA: El elemento debe ser soportado por el LMS ya que el elemento es obligatorio <p>Ejemplo de uso del SCO: Los SCOs pueden usar llamadas al cmi.core._children para determinar si un cierto elemento está implementado por el LMS.</p> <pre>var coreChildren = LMSGetValue("cmi.core._children"); if (coreChildren.indexOf("student_name") != -1) { studentName = LMSGetValue("cmi.core.student_name"); }</pre>
cmi.core.student_id	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio para el LMS: Sí</p> <p>Tipo de datos: CMIIentifier</p> <p>Accesibilidad al SCO: Solo lectura</p>	<p>Definición: Un único identificador alfa-numérico que representa un único usuario del sistema LMS.</p> <p>Uso: Usado para identificar un estudiante.</p> <p>Formato: Hasta 255 caracteres alfa-numéricos sin espacios. Guiones y comas están permitidos. Los puntos no están permitidos. No hay diferencia entre mayúsculas y minúsculas..</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> • Inicialización: El LMS es responsable, basado en el registro del estudiante • LMSGetValue(): Devuelve el valor actual almacenado por el LMS al estudiante. <ul style="list-style-type: none"> ○ Ejemplo de valor devuelto: "Joe_Student1" "JS-2000" "joe2000-3" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. • LMSSetValue(): El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error:

<p>Obligatorio LMS: sí</p> <p>Tipo de datos: CMISString255</p> <p>Accesibilidad SCO: Lectura/Escritura</p>	<p>Formato: La implementación depende. El LMS simplemente almacena la información y la devuelve al SCO cuando el estudiante está volviendo al aprendizaje si el SCO pregunta por ello. El formato siempre es correcto ya que es el SCO quien manda la información a almacenar y quien la recibe y por tanto eso no pasa por cuenta del LMS. Si el alumno accede por primera vez al SCO, lesson_location puede ser igual a una cadena vacía.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ● Inicialización: El LMS debería introducir una cadena vacía en esta variable. Sin embargo el SCO puede modificar este valor y recuperarlo posteriormente. ● LMSGetValue(): Devuelve el valor actual almacenado de dicha variable <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. ● LMSSetValue(): Actualiza el valor de la variable al valor indicado. El valor debe coincidir con el tipo de dato para este elemento <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. <p>Ejemplo de uso del SCO: El SCO puede usar lesson_location como un marcador: -Al lanzar el SCO, el SCO puede posicionar al estudiante donde el estudiante lo dejó en una sesión anterior - Al salir del SCO, el SCO puede almacenar la posición del estudiante para que continúe en la misma posición la próxima vez que vuelva a entrar.</p> <p>// Ejemplo de SCO escrito en Java Script y HTML // Esta función podría existir en una función llamada onLoad() que se ejecutaría cuando la página HTML se carga</p> <pre>var coreSCOLocation = LMSGetValue("cmi.core.lesson_location"); if (LMSGetLastError() == "0") { // coreSCOLocation contiene un enlace definido en la página HTML // Empezar la lección donde el estudiante la dejó window.location.hash = coreSCOLocation; } else { // Procesamiento de errores }</pre>
cmi.core.credit	
<p>Llamadas al API admitidas: LMSGetValue()</p> <p>Obligatorio LMS: Sí</p> <p>Tipo de dato: CMIVocabulary (Credit) "credit" "no-credit"</p>	<p>Definición: Indica si el estudiante está siendo evaluado por el LMS basado en la puntuación o en forma de aprobado/suspenseo.</p> <p>Uso: Usado por el LMS para indicar si el alumno está haciendo el curso para ser puntuado.</p> <p>cmi.core.credit se utiliza junto con lesson_mode y también junto con lesson_status</p> <p>Formato: Hay dos valores posibles:</p>

<p>accesibilidad del SCO: Sólo lectura</p>	<ul style="list-style-type: none"> ● “credit”. Significa que el estudiante va a ser evaluado. El LMS le dice al SCO que los datos que el SCO manda al LMS son para ser evaluados. ● “no-credit”. Significa que el estudiante no va a ser evaluado. El LMS le dice al SCO que los datos que el SCO manda al LMS no los utiliza para evaluar al estudiante. <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ● Inicialización: El LMS es responsable de determinar si el estudiante está haciendo el curso para ser evaluado o para no serlo. ● LMSGetValue(): Devuelve el valor actual de la variable almacenada por el LMS. <ul style="list-style-type: none"> ○ Ejemplo de valores devueltos: "no-credit" "credit" ○ Códigos de error: <ul style="list-style-type: none"> ● 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. ● LMSSetValue(): El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ■ 403 Los elementos son de sólo lectura . Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que el elemento es obligatorio) y se hace una llamada a LMSSetValue() en este elemento, entonces el LMS debería actualizar el código de error a 403. ■ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. <p>Ejemplo de uso del SCO: El SCO podría usar el valor devuelto desde el LMS para determinar lo que se muestra en el explorador. Puede haber diferentes contenidos en el SCO que es mostrado en el explorador según sea para evaluarlo o no evaluarlo.</p> <pre>var creditFlag = LMSGetValue("cmi.core.credit") if (creditFlag == "credit") { // El estudiante está haciendo el curso para ser evaluado. } else { // El estudiante está haciendo el curso para no ser evaluado. }</pre>
cmi.core.lesson_status	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: Sí</p> <p>Tipo de datos: CMIVocabulary (estado)</p> <p>passed completed failed incomplete browsed not attempted</p>	<p>Definición: Este es el estado actual del estudiante tal y como es determinado por el LMS. Hay 6 valores de estado posibles.</p> <p>Uso: Normalmente el SCO determina su propio estado y lo pasa al LMS. Si en cmi.core.credit se almacena el valor credit y el manifiesto del SCO tiene la capacidad de evaluar el LMS puede cambiar el estado a “aprobado” o “suspense” dependiendo de la puntuación del estudiante y comparada con las instrucciones del SCO.</p> <ol style="list-style-type: none"> 1) Si el SCO no tiene la capacidad de evaluar el LMS no tiene la capacidad de sustituirle. 2) Si el estudiante realiza el SCO para no ser evaluado no hay cambio en la variable lesson_status con una excepción: si lesson_mode está en “browse”, la variable “lesson_status” puede cambiar a “browsed” incluso si cmi.core.credit está en non-credit.

<p>Accesibilidad del SCO: Leer/Escribir</p>	<p>3) Al volver a entrar en el SCO, el LMS puede cambiar de estado a “aprobado”, “suspense” o “visto”. “Aprobado” y “suspense” están definidos según los criterios del SCO. Se cambiará de estado a “visto” cuando el SCO fué lanzado inicialmente en estado “pendiente de ver”.</p> <p>Formato: Hay 6 posibles valores (los valores del vocabulario deben ser en inglés para que sea totalmente compatible con todo tipo de cursos:</p> <ul style="list-style-type: none"> ● passed: El número necesario de objetivos en el SCO han sido aprobados o se logró la puntuación para aprobar. ● completed: El SCO puede o no puede ser aprobado pero todos sus contenidos pueden haber sido vistos por el estudiante. Esto es lo que indica la variable completed. ● failed: El estudiante ha suspendido los contenidos del SCO. En cambio esto no obliga a que todos los contenidos hayan sido vistos. ● incomplete: El SCO ha sido empezado pero no terminado ● browsed: El estudiante ya ha lanzado el LMS antes. ● not attempted: significa que el estudiante hizo un intento de cargar el curso pero por alguna razón el curso ni siquiera ha sido empezado. Quizás el alumno tan sólo ha leído el árbol de contenidos y decidió que no estaba preparado para afrontar el curso. Cualquier algoritmo del SCO puede ser usado para cambiar el valor “not attempted” a “incomplete” <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ● Inicialización: Si es la primera vez que el nuevo estudiante accede al SCO, la variable lesson_status está situada en el valor “not attempted”. El LMS es responsable de situar el valor inicial en “not attempted” <ul style="list-style-type: none"> ○ Otros comportamientos obligatorios para el LMS: si un SCO almacena un valor en cmi.core.lesson_status entonces no hay ningún problema. Sin embargo, SCORM no obliga al SCO a establecer la variable cmi.core.lesson_status. Por tanto hay algunos requerimientos que debe satisfacer el LMS para estos casos: <ul style="list-style-type: none"> ▪ En el lanzamiento inicial el LMS debería establecer el valor “not attempted” en cmi.core.lesson_status. ▪ Cuando se ejecuta LMSFinish() o el usuario sale de la aplicación el LMS debe establecer cmi.core.lesson_status en “completed”. ▪ Una vez establecido el valor de cmi.core.lesson_status en “completed”, el LMS debe mirar si el SCO tiene activo el módulo de evaluación comprobando la variable cmi.student_data.mastery_score, si es admitida, o el manifiesto del SCO. Si se suministra un módulo de evaluación y el SCO estableció valores en cmi.core.score.raw el LMS comparará el módulo de evaluación con cmi.core.score.raw y almacenará en cmi.core.lesson_status “passed” o “failed” según haya aprobado o suspendido respectivamente. Si no se suministra módulo de evaluación el Lms dejará cmi.core.lesson_status como “completed”, es decir, terminado. ● LMSGetValue(): Devuelve el valor almacenado en el modelo de datos. Debe ser una del conjunto de variables definidas. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. ● LMSSetValue(): Actualiza el valor de la variable al valor indicado. El valor debe coincidir con el tipo de dato para este elemento <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de
--	--

	<p>error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio.</p> <ul style="list-style-type: none"> • Ejemplo de asignación y recuperación de variables:: "completed" "failed" "browsed" <p>Uso del SCO:</p> <ul style="list-style-type: none"> ▪ passed – usado cuando el SCO se usa para evaluar. ▪ failed – usado cuando el SCO se usa para evaluar. ▪ completed – usado cuando el SCO no se usa para evaluar. ▪ incomplete – usado cuando se abandona la aplicación antes de terminarla, ya sea evaluando o no evaluando. ▪ browsed – usado cuando el lesson_mode sea "browse" ▪ not attempted – El SCO nunca debería introducir este valor en lesson_status (es el valor inicial que asigna el LMS cuando el estudiante accede por primera vez al SCO). <p>Ejemplo del uso del SCO:</p> <pre>var lessonStatus = LMSGetValue("cmi.core.lesson_status"); if (lessonStatus == "failed") { // El estudiante suspendió el SCO, actuar en consecuencia. } else { // El estudiante aprobó el SCO, actuar en consecuencia. }</pre>
cmi.core.entry	
<p>Llamadas admitidas al API LMSGetValue()</p> <p>Obligatorio LMS: Yes</p> <p>Tipo de datos: CMIVocabulary (Entry)</p> <p>"ab-inicio" "resume" "" – cadena vacía</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Definición: Indicación de si el estudiante ha accedido al SCO con anterioridad.</p> <p>Uso: Cuando un estudiante carga el SCO por primera vez, cmi.core.entry debería establecerse en "AB-inicio" por el LMS. Si un estudiante vuelve a entrar en un contenido suspendido. Se establecería el valor "resume" en dicha variable.</p> <p>Formato: Tres valores posibles:</p> <ul style="list-style-type: none"> • "ab-inicio": Esto indica que es la primera vez que un estudiante entra al SCO. Puesto que un estudiante podría haber pasado todos los objetivos de un SCO haciendo un pre-test, el valor "not attempted en lesson_status no es un indicador fiable. Esto significaría que un SCO podría ser aprobado sin que el estudiante ni siquiera lo haya visto antes. • "resume": Indica que el estudiante ha estado en el SCO con anterioridad. El estudiante está empezando de nuevo un SCO suspendido. • "": La cadena vacía se usa para representar una entrada del estudiante en el SCO que no es ninguna de las anteriores. <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: En el lanzamiento inicial del LMS se debe inicializar el modelo de datos con "ab-inicio". <ul style="list-style-type: none"> ○ Comportamiento posterior: Al recibir LMSFinish() o cuando el estudiante se sale de la aplicación, el LMS debe establecer cmi.core.entry en "" – cadena vacía- o "resume". Esto se determina por el LMS analizando la variable cmi.core.exit. Si vale "suspend" se pondrá en cmi.core.entry en "resume" hasta la próxima vez que se inicie la aplicación. Si hay otro valor o no hay valor en cmi.core.exit , cmi.core.entry debe establecerse en "" –cadena vacía-. ▪ LMSGetValue(): Devuelve el valor almacenado, el cual debe ser una palabras del conjunto de palabras admitidas. <ul style="list-style-type: none"> ○ Ejemplo de valores devueltos: "AB-inicio" "resume" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de

	<p>error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio.</p> <ul style="list-style-type: none"> • LMSSetValue(): El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 403 Los elementos son de sólo lectura . Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que el elemento es obligatorio) y se hace una llamada a LMSSetValue() en este elemento, entonces el LMS debería actualizar el código de error a 403. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. <p>Ejemplo del uso del SCO:</p> <pre>var entryStatus = LMSGetValue("cmi.core.entry") if (LMSGetLastError() == "0") { if (entryStatus == "resume") { // El estudiante está volviendo a empezar el SCO } else { // Esta es la primera vez que el estudiante se introduce en el SCO } } else { // Hay un error, actuar consecuentemente }</pre>
<p>cmi.core.score Indica el rendimiento del estudiante Children de cmi.core.score: raw, min, max</p>	
<p>cmi.core.score._children</p>	
<p>Llamadas soportadas por el API LMSGetValue()</p> <p>Obligatorio LMS: Sí</p> <p>Tipo de datos: CMISString255</p> <p>Accesibilidad del SCO: Sólo lectura</p>	<p>Definición: La palabra clave _children se usa para determinar todos los elementos de la categoría de puntuación que son soportados por el LMS. Si un elemento no tiene children, pero los admite, se devuelve una cadena vacía. Si un elemento no es admitido no se devuelve ningún valor. Una solicitud del último error puede verificar que el elemento no es admitido.</p> <p>Uso: se usa para determinar los elementos (children) de cmi.core.score que son admitidos por el LMS. Raw es el único elemento obligatorio que debe ser admitido.</p> <p>Formato: El valor devuelto es una lista separada por comas de todos los nombres de la categoría de puntuación que son admitidos por el LMS.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El conjunto de children admitidos para este grupo. Por tanto con una llamada a LMSGetValue(), la lista apropiada de elementos admitidos es devuelta. ▪ LMSGetValue(): Devuelve una lista separadas por comas de los elementos admitidos. <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.core.score._children") ○ Ejemplo de valores devueltos: "raw" – <i>el LMS debe admitir al menos este elemento</i> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error de no implementación. Si el elemento cmi.core._children no es soportado por el LMS se devolvería una cadena vacía. NOTA: el cmi.core._children debe ser admitido por el LMS ya que

	<p>el elemento es obligatorio.</p> <ul style="list-style-type: none"> • LMSSetValue(): El LMS debería establecer un código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 402 – Valor establecido inválido: el elemento es una palabra clave. Si el elemento es admitido por el LMS (el elemento debe ser admitido ya que el elemento es obligatorio y una llamada a LMSSetValue sobre este elemento debe colocar el código de error en 402. ▪ 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. NOTA: El elemento debe ser soportado por el LMS ya que el elemento es obligatorio <p>Ejemplo de uso del SCO:</p> <pre>var scoreChildren = LMSGetValue("cmi.core.score._children"); if (coreChildren.indexOf("min") != -1) { LMSSetValue("cmi.core.score.min", "10"); }</pre>
cmi.core.score.raw	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: Sí</p> <p>Tipo de datos: CMIDecimal o CMIBlank</p> <p>Accesibilidad SCO: Lectura/Escritura</p>	<p>Definición: Indicación de la actuación del estudiante durante su último intento en el SCO. Esta puntuación puede ser determinada y calculada de cualquier manera que tenga sentido para el diseñador del SCO. Por ejemplo, podría reflejar el porcentaje de objetivos completado, podría ser el resultado bruto en un test con múltiples posibilidades.</p> <p>El cmi.core.score.raw debe ser un valor normalizado entre 0 y 100.</p> <p>Uso: Cuando el estudiante carga el SCO por primera vez, el cmi.core.score.raw debe estar en cadena vacía. Las siguientes veces cmi.core.score.raw refleja lo que fue grabado en la sesión previa del estudiante. Si el SCO no tiene valor en cmi.core.score.raw se debe devolver una cadena vacía.</p> <p>Formato: número decimal o en blanco.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS debe inicializar esto con una cadena vacía. El SCO es el responsable de establecer este valor. Si LMSGetValue() se ejecuta antes de que el SCO haya establecido este valor, entonces el LMS debería devolver una cadena vacía. ▪ LMSGetValue(): Devuelve el valor almacenado en el modelo de datos. <ul style="list-style-type: none"> ○ Llamada de ejemplo del API: LMSGetValue("cmi.core.score.raw") <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. • LMSSetValue(): Actualiza el valor de la variable al valor indicado. El valor debe coincidir con el tipo de dato para este elemento <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. <p>▪ Ejemplo sobre valores devueltos: "90"</p>

	<p>"85.7" ""</p> <p>Ejemplo sobre el uso del SCO: El SCO podría usarse para seguir el resultado neto del estudiante en el SCO.</p> <p>LMSSetValue("cmi.core.score.raw","85");</p>
cmi.core.score.max	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMIDecimal or CMIBlack</p> <p>Accesibilidad SCO: Leer/Escribir</p>	<p>"" Definición: La máxima puntuación o el número total de aciertos que el estudiante podría haber logrado.</p> <p>El cmi.core.max debe ser un valor normalizado entre 0 y 100.</p> <p>Uso: indica la máxima puntuación que el estudiante podría haber alcanzado.</p> <p>Formato: número decimal o en blanco.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS debería inicializarlo con una cadena vacía ("") en el lanzamiento inicial del SCO. El SCO es responsable de tratar este valor. Si se ejecuta un LMSGetValue() antes de que el SCO haya almacenado algo en esta variable entonces el LMS debería devolver una cadena vacía. ▪ LMSGetValue(): Devuelve el valor almacenado en el modelo de datos. El valor devuelto debe ser de tipo CMIDecimal o CMIBlack. <ul style="list-style-type: none"> ○ Ejemplo de una llamada del API: LMSGetValue("cmi.core.score.max") ○ Código de error: 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. NOTA: El elemento debe ser soportado por el LMS ya que el elemento es obligatorio ▪ LMSSetValue(): Situa el elemento del modelo de datos en el valor dado. Éste debe corresponder al tipo de datos para este elemento. <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.core.score.max","100") ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de valores devueltos o almacenados: "100" "5" <p>Ejemplo de uso del SCO: <pre>var scoreChildren = LMSGetValue("cmi.core.score._children"); if (coreChildren.indexOf("max") != -1) { LMSSetValue("cmi.core.score.max","100"); }</pre> </p>
cmi.core.score.min	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMIDecimal or CMIBlack</p>	<p>Definición: La mínima puntuación o el número total de aciertos que el estudiante podría haber logrado.</p> <p>El cmi.core.max debe ser un valor normalizado entre 0 y 100.</p> <p>Uso: indica la mínima puntuación que el estudiante podría haber alcanzado.</p> <p>Formato: número decimal o en blanco.</p> <p>Comportamiento del LMS:</p>

<p>Accesibilidad SCO: Leer/Escribir</p>	<ul style="list-style-type: none"> ▪ Inicialización: El LMS debería inicializarlo con una cadena vacía ("") en el lanzamiento inicial del SCO. El SCO es responsable de tratar este valor. Si se ejecuta un LMSGetValue() antes de que el SCO haya almacenado algo en esta variable entonces el LMS debería devolver una cadena vacía. ▪ LMSGetValue(): Devuelve el valor almacenado en el modelo de datos. El valor devuelto debe ser de tipo CMIDecimal o CMIBlank. <ul style="list-style-type: none"> ○ Ejemplo de una llamada del API: LMSGetValue("cmi.core.score.max") ○ Código de error: 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. ▪ LMSSetValue(): Situa el elemento del modelo de datos en el valor dado. Éste debe corresponder al tipo de datos para este elemento. <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.core.score.max","100") ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de valores devueltos o almacenados: "100" "5" <p>Ejemplo del uso del SCO:</p> <pre>var scoreChildren = LMSGetValue("cmi.core.score._children"); if (coreChildren.indexOf("min") != -1) { LMSSetValue("cmi.core.score.min","10"); }</pre>
cmi.core.total_time	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: Sí</p> <p>Tipo de datos: CMITimespan</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Definición: Tiempo acumulado de todas las sesiones de un estudiante en el SCO.</p> <p>Uso: Usado para realizar el seguimiento del tiempo total que ha pasado un alumno con un SCO. El LMS debe inicializarlo en un valor por defecto cuando el SCO es lanzado por primera vez y entonces usar los valores descritos por el SCO (session_time) para llevar una cuenta acumulada.</p> <p>Formato: Horas, minutos y segundos separados por dos puntos. HHHH:MM:SS.SS Horas tiene un mínimo de 2 dígitos y un máximo de 4 dígitos. Los minutos consistirán en 2 dígitos exactos. Los segundos tendrán 2 dígitos con un punto decimal opcional para indicar décimas o centésimas.</p> <p>Coportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS debe inicializarlo en "0000:00:00.00" en el primer lanzamiento del SCO. <ul style="list-style-type: none"> ○ Comportamiento adicional: Un SCO es capaz, en una sola ejecución, de llevar múltiples almacenamientos del cmi.core.total_time. Cuando el SCO ejecuta LMSFinish() o el usuario sale del sistema, el LMS debería coger el último cmi.core.session_time que el SCO almacenó y acumularlo en cmi.core.total_time. En el siguiente lanzamiento del SCO, una llamada a LMSGetValue() para la variable cmi.core.total_time, el LMS debería devolver el tiempo total acumulado. Los Lms no deben acumular los tiempos múltiples enviados por los SCO con llamadas a LMSSetValue() para la variable cmi.core.session_time. Si se hacen llamadas múltiples a LMSSetValue() por cmi.core.session_time, el LMS debe sobrescribir cualquier valor existente de esa variable.

	<ul style="list-style-type: none"> ▪ LMSGetValue(): Devuelve el valor almacenado en el modelo de datos. El valor devuelto debe de ser de tipo CMITimespan. <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.core.total_time") ○ Ejemplo de valores devueltos: "00:29:00" "01:27:45.5" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. • LMSSetValue():El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 403 Los elementos son de sólo lectura . Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que el elemento es obligatorio) y se hace una llamada a LMSSetValue() en este elemento, entonces el LMS debería actualizar el código de error a 403. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: los elementos deben ser admitidos por el LMS dado que el elemento es obligatorio. <p>Ejemplo de uso del SCO:</p> <pre>var totalTime = LMSGetValue("cmi.core.total_time"); if (LMSGetLastError() == "0") { // Uso de cmi.core.total_time }</pre>
cmi.core.lesson_mode	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMIVocabulary (Mode)</p> <p>"browse" "normal" "review"</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Definición: Identifica el comportamiento deseado del SCO después del lanzamiento de este. Muchos SCO tienen un único comportamiento. Sin embargo, algunos SCOs pueden presentar distintas cantidades de información o presentarla en distinto orden reflejando las distintas formas de aprendizaje basadas en los deseos del diseñador del SCO. Los diseñadores pueden hacer que un SCO tenga comportamientos distintos ilimitados. Este estándar admite la comunicación de tres parámetros que pueden resultar en comportamientos distintos del SCO.</p> <p>Uso: Se usa para representar los diferentes modos en que un SCO puede ser lanzado. Se usa junto con lesson_status.</p> <p>Formato: Tres valores posibles:</p> <ul style="list-style-type: none"> • "browse": El estudiante quiere hacer una vista previa pero no necesariamente ser evaluado. • "normal": indica que el SCO se debe comportar de forma que el alumno sea evaluado. • "review": El estudiante ya ha visto el contenido al menos una vez y ha sido evaluado. <p>Si lesson_mode no es reconocido o hay errores entonces el SCO asume el estado "normal" por defecto.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS debería determinar el modo en que el SCO está siendo lanzado. Nota: ahora mismo en el modelo SCORM 1.2 no existe forma de saber si un contenido de aprendizaje puede ser lanzado de diferentes maneras. ▪ LMSGetValue(): Devuelve el valor almacenado en el modelo de datos.

	<ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.core.lesson_mode") ○ Ejemplo de valores devueltos: "browse" "normal" "review" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ● LMSSetValue():El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 403 Los elementos son de sólo lectura . Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que el elemento es obligatorio) y se hace una llamada a LMSSetValue() en este elemento, entonces el LMS debería actualizar el código de error a 403. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO:</p> <pre>var mode = LMSGetValue("cmi.core.lesson_mode"); if (LMSGetLastError() == "0") { if (mode == "browse") { // El estudiante está en modo "browse" } else if (mode == "review") { //El estudiante ya ha visto el contenido y ha sido evaluado. Debe estar revisando. } else { // El estudiante está lanzando el SCO en modo normal } }</pre>
cmi.core.exit	
<p>Llamadas admitidas por el API: LMSSetValue()</p> <p>Obligatorio LMS: sí</p> <p>Tipo de datos: CMIVocabulary (Exit) "time-out" "suspend" "logout" "" - empty string</p> <p>Accesibilidad SCO: Sólo escritura</p>	<p>Definición: una indicación de como o porqué el estudiante abandonó el SCO.</p> <p>Uso: Se usa para indicar la razón por la que el estudiante abandonó la aplicación.</p> <p>Formato: Tres valores posibles:</p> <ul style="list-style-type: none"> ● "time-out": Indica que el SCO finalizó la aplicación porque ha pasado un tiempo excesivo o el tiempo máximo dado ha sido excedido. Este tiempo se puede encontrar en el manifiesto(adlcp:maxtimeallowed). ● "suspend": Indica que el estudiante sale del SCO con intención de continuar más tarde en el mismo punto donde lo dejó. ● "logout": Indica que el estudiante salió de la aplicación desde dentro del SCO en lugar de volver al LMS para salir. Esto implica que el SCO pasó el control al LMS y el LMS automáticamente retiró al estudiante de la aplicación – después de actualizar los valores correspondientes en el modelo de datos- ● "" : Se usa para representar un estado normal de salida. <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El elemento no necesita ser inicializado. Nunca se hace un LMSGetValue sobre este elemento. El elemento se controla con el SCO. ○ Comportamiento adicional: Un SCO tiene la opción de establecer cmi.core.exit en uno de los cuatro valores. Basado en

	<p>dicho valor, el comportamiento del LMS debería ser como sigue:</p> <ul style="list-style-type: none"> ▪ Si el SCO establece cmi.core.exit en "time-out", el LMS debería establecer cmi.core.entry en "" (cadena vacía) en el siguiente lanzamiento del SCO. ▪ Si el SCO establece cmi.core.exit en "suspend", el LMS debería establecer cmi.core.entry en "resume" en el siguiente lanzamiento del SCO. ▪ Si el SCO establece cmi.core.exit en "logout", el LMS debería establecer cmi.core.entry en ""(cadena vacía) la próxima vez que se lance el SCO. Además, el LMS debería sacar al estudiante del curso cuando el SCO que estableció cmi.core.exit en "logout" haya ejecutado LMSFinish() o el usuario abandone la aplicación. ▪ Si el SCO establece cmi.core.exit en ""(cadena vacía), el LMS debe establecer cmi.core.entry en "" la próxima vez que el SCO sea lanzado. ▪ Si el SCO no estableció cmi.core.exit en ningún valor entonces el LMS debería establecer cmi.core.entry en ""(cadena vacía) la próxima vez que se lance el SCO. <ul style="list-style-type: none"> ▪ LMSGetValue(): El LMS debería establecer un código de error de acuerdo con lo siguiente y devolver una cadena vacía. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. NOTA: el elemento debe ser admitido por el LMS ya que es obligatorio. ▪ 404 – El elemento es de solo escritura. Si un SCO intenta llamar a LMSGetValue() en este elemento, el LMS debe establecer el código de error en 404 y devolver una cadena vacía. ▪ LMSSetValue(): Establece nuevos valores en un elemento del modelo de datos. El valor debe coincidir con el tipo de dato de ese elemento: <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.core.exit","logout") ○ Ejemplo de valores establecidos: "time-out" "suspend" "logout" ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: el elemento debe ser admitido por el LMS ya que es obligatorio. <p>Ejemplo del uso del SCO: LMSSetValue("cmi.core.exit","time-out")</p>
cmi.core.session_time	
<p>Llamadas admitidas por el API: LMSSetValue()</p> <p>Obligatorio LMS: sí</p> <p>Tipo de dato: CMITimespan</p> <p>Accesibilidad SCO: Sólo escritura</p>	<p>Definición: Es la cantidad de horas, minutos y segundos que el estudiante ha pasado con el SCO en el momento en que lo abandona, es decir, el tiempo de la sesión en un sólo uso del SCO.</p> <p>Uso: Se usa para seguir el tiempo de la sesión de un alumno. El LMS usa este tiempo para determinar el cmi.core.total_time.</p> <p>Formato: HHHH:MM::SS.SS Horas, minutos y segundos separados por dos puntos. HHHH:MM:SS.SS Horas tiene un mínimo de 2 dígitos y un máximo de 4 dígitos. Los minutos consistirán en 2 dígitos exactos. Los segundos tendrán 2 dígitos con un punto decimal opcional para indicar décimas o centésimas.</p>

<p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El elemento no necesita ser inicializado por el LMS. Nunca hay una llamada con LMSGetValue() a este elemento. <ul style="list-style-type: none"> ○ Comportamiento adicional: Un SCO es capaz de llevar de una vez múltiples establecimientos de cmi.core.session_time. Cuando se ejecuta LMSFinish() o el usuario abandona la aplicación, el LMS debe acumular el cmi.core.session_time al cmi.core.total_time. El LMS no debería acumular múltiples tiempos enviados al LMS por llamadas a LMSSetValue() sobre cmi.core.session_time. Si se hiciera, el LMS debería sobrescribir cualquier valor existente. ▪ LMSGetValue(): El LMS debería establecer un código de error de acuerdo con lo siguiente y devolver una cadena vacía. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. NOTA: el elemento debe ser admitido por el LMS ya que es obligatorio. ▪ 404 – El elemento es de solo escritura. Si un SCO intenta llamar a LMSGetValue() en este elemento, el LMS debe establecer el código de error en 404 y devolver una cadena vacía. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.core.session_time","0010:34:34.56") ○ Ejemplo de valores devueltos: "0010:34:34.56" "05:15:00" ○ Código de error: <ul style="list-style-type: none"> ▪ 205 – Tipo de dato incorrecto: Si un LMSSetValue() falla al no coincidir el tipo de dato de la variable que lleva con el que debería tener. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. NOTA: el elemento debe ser admitido por el LMS ya que es obligatorio. <p>Ejemplo del uso del SCO: LMSSetValue("cmi.core.session_time","0000:12:30")</p>

cmi.suspend_data	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: sí</p> <p>Tipo de datos: CMISString4096</p> <p>Accesibilidad del SCO: Leer / Escribir</p>	<p>Definición: La única información generada por el SCO durante usos previos que se necesita para el uso actual. Ésta única información se aplica cuando se inicia el SCO. Normalmente este es el elemento que utiliza el SCO para reiniciar información. Estos datos normalmente se crean por el SCO y se almacenan en el LMS para volver al SCO la siguiente vez que éste sea lanzado.</p> <p>El LMS debe tener espacio aparte para este grupo de variables por cada SCO y por cada estudiante mientras este siga en el curso.</p> <p>Uso: Sólo disponible al reinicio del SCO. El SCO puede establecer este valor si un estudiante existe antes de que el SCO sea completado.</p> <p>Formato: La única limitación para estos datos son:</p> <ol style="list-style-type: none"> 1. Los datos deben ser transferidos en formato ASCII. El SCO puede entonces convertirlo en cualquier formato que necesite. 2. Para evitar una sobrecarga del sistema, el LMS debería limitar este elemento a 4096 bytes de datos.

	<p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS debería inicializar este elemento en una cadena vacía. El SCO es responsable de establecer valores en este elemento. ▪ LMSGetValue(): Devuelve el valor almacenado en el modelo de datos. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. NOTA: el elemento debe ser admitido por el LMS ya que es obligatorio. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: el elemento debe ser admitido por el LMS ya que es obligatorio. <p>Ejemplo de uso del SCO: // Bajo el relanzamiento del SCO por un estudiante. // Obtener el Suspend Data var suspendData = LMSGetValue("cmi.suspend_data") if (LMSGetLastError() == "0") { // Uso de suspend data }</p>
--	--

cmi.launch_data	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: Sí</p> <p>Tipo de datos: CMISString4096</p> <p>Accesibilidad del SCO: Sólo lectura</p>	<p>Definición: Una única información generada en la creación del SCO que se necesita para todo uso. Sin esta información el SCO podría no ejecutarse.</p> <p>Uso: cmi.launch_data está disponible para el SCO para ayudar en su lanzamiento. Siempre será igual para un SCO dado.</p> <p>Formato: campo de texto. Esto contiene cualquier sistema único de información necesario para el SCO para funcionar bien. Esto está limitado a 4096 caracteres.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Este valor debería ser inicializado por el LMS utilizando el manifiesto (la parte de adlcp:datafromIMS). Si no se encuentran datos de lanzamiento en el manifiesto, entonces estos deben ser establecidos como como una cadena vacía "". ▪ LMSGetValue(): Devuelve el valor almacenado en el modelo de datos. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. • LMSSetValue(): El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error:

	<ul style="list-style-type: none"> ▪ 403 Los elementos son de sólo lectura . Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que el elemento es obligatorio) y se hace una llamada a <code>LMSSetValue()</code> en este elemento, entonces el LMS debería actualizar el código de error a 403. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO:</p> <pre>var launchData = LMSGetValue("cmi.launch_data") if (LMSGetLastError() == "0") { // uso de launch_data }</pre>
--	--

<p>cmi.comments Mecanismo para incluir y distribuir comentarios para un SCO</p> <p>cmi.comments – representa la habilidad del SCO de enviar/recibir comentarios. cmi.comments_from_lms – representa la habilidad de suministrar comentarios al SCO (sólo lectura desde la perspectiva del SCO)</p>	
<p>cmi.comments</p>	
<p>Llamadas soportadas al API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMISString4096</p> <p>Accesibilidad SCO: Leer/Escribir</p>	<p>Definición: formulario libre de retroalimentación del SCO. Por ejemplo, el estudiante puede tener la opción de dejar comentarios en cualquier punto del SCO o se le puede pedir comentarios al final del SCO. El comentario puede tener también una indicación en el SCO de donde o cuando fue creado. Un lugar también se puede incluir en el comentario.</p> <p>Uso: Se usa para permitir al SCO enviar comentarios al LMS sobre el SCO. También se podría usar para permitir a los estudiantes introducir comentarios.</p> <p>Formato: Formulario libre con texto alfanumérico y caracteres especiales.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Ninguna, deberís ser inicializado con una cadena vacía. El SCO es responsable de establecer este valor. Si se hace <code>LMSGetValue()</code> antes de que el SCO haya establecido el valor de este elemento se debe devolver una cadena vacía. ▪ LMSGetValue(): <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. NOTA: el elemento debe ser admitido por el LMS ya que es obligatorio. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a <code>LMSSetValue()</code> con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. NOTA: el elemento debe ser admitido por el LMS ya que es obligatorio. ▪ Ejemplo de establecimiento y devolución de valores: Depende de la implementación del SCO.

	<p>Ejemplo de uso del SCO</p> <pre>var comments = LMSGetValue("cmi.comments") if (LMSGetLastError() == "0") { // uso de launch_data }</pre>
cmi.comments_from_lms	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMISString4096</p> <p>Accesibilidad del SCO: Sólo lectura</p>	<p>Definición: Este elemento representa comentarios del LMS. Estos comentarios son presentados por el SCO al estudiante cuando es apropiado.</p> <p>Uso: Se usa para permitir al SCO ver comentarios relacionados con él y originados en el LMS.</p> <p>Formato: no hay formato específico.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Estos comentarios deberían ser inicializados por el LMS si los hay. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.comments_from_lms") ○ Ejemplo de valores devueltos: Depende de la implementación ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. • LMSSetValue(): El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 403 Los elementos son de sólo lectura . Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que el elemento es obligatorio) y se hace una llamada a LMSSetValue() en este elemento, entonces el LMS debería actualizar el código de error a 403. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO:</p> <pre>// El SCO puede querer ver si hay comentarios del LMS para el estudiante var commentsFromLMS = LMSGetValue("cmi.comments_from_lms"); if (LMSGetLastError() == "0") { //uso de los comentarios desde el LMS</pre>

cmi.objectives	
Identifica si el estudiante a alcanzado los objetivos cubiertos en el SCO	
Children of cmi.objectives:	
id, score, status	
cmi.objectives_children	
Llamadas admitidas por	Definición: Las palabras claves "children" se usan para determinar todos los elementos

<p>el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMISString255</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>incluidos en la categoría cmi.objectives que son admitidos por el LMS. Si un elemento no tiene "children", pero los admite, se devuelve una cadena vacía. Si no es admitido no se devuelve nada. Se puede concluir que no se admite haciendo una llamada al último error.</p> <p>Uso: Se usa para determinar qué elementos son admitidos por el LMS</p> <p>Formato: El valor devuelto es una lista separada por comas de todos los nombres de los elementos. Identifica que información detallada se puede recoger en escenarios complicados, por ejemplo, simulaciones.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Conjunto de valores admitidos para este grupo. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.objectives._children") ○ Ejemplo de valores devueltos: "id,score" "id,status" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Códigos de error: <ul style="list-style-type: none"> ▪ 402 – Valor establecido no válido, el elemento es una palabra clave. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo del uso del SCO: El SCO podría usar este elemento para determinar qué elementos son admitidos por el LMS.</p> <pre>var objChildren = LMSGetValue("cmi.objectives._children") if (objChildren.indexOf("status") != -1) { // Establecer los estados objetivos }</pre>
cmi.objectives._count	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMInteger</p> <p>Accesibilidad del SCO: Sólo lectura</p>	<p>Definición: La palabra clave _count se usa para determinar el número actual de registros en la lista cmi.objectives. Se devuelve el número total de entradas. Si el SCO no sabe el número de registros de cmi.objectives, puede comenzar la cuenta del estudiante con cero. Esto sobrescribiría cualquier información sobre objetivos almacenada en la primera posición del índice. Que se sobrescriba o que se anexe es una decisión hecha por el autor del SCO cuando este se crea.</p> <p>Uso: Se usa para determinar el número de objetivos almacenados por el LMS. Los SCO's podrían usar este número para determinar qué registro objetivo recuperar. Si se devuelve "3" entonces el SCO sabe que puede referenciar registros del 0 al 2.</p> <p>Formato: Devuelve el valor como un integer (número entero) que indica el número de objetos que hay actualmente en una lista de elementos o un array.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS es responsable de inicializar esto a cero en el lanzamiento inicial del SCO cuando ningún dato de objetos ha sido insertado por el SCO. ▪ LMSGetValue(): Devuelve el número total de entradas. <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.objectives._count") ○ Ejemplo de valores devueltos: "0" "4" ○ Códigos de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error

	<p>se actualiza indicando que el elemento no es admitido.</p> <ul style="list-style-type: none"> ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Error Code: <ul style="list-style-type: none"> ▪ 402 – Establecimiento de valores inválido, el elemento es una palabra clave ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo del uso del SCO: El SCO podría usar el elemento <code>_count</code> para determinar qué array usar en procesos posteriores.</p> <pre>//conseguir la cuenta de objetos registrados por el LMS var totalObj = LMSGetValue("cmi.objectives._count") //sustraer uno del total (para compensar el array basado en cero) var request = "cmi.objectives." + totalObj - 1 + ".id" // Llamada a nuevo Objective ID LMSSetValue(request, "Obj1_110")</pre>
cmi.objectives.n.id	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMIIentifier</p> <p>Accesibilidad SCO: Leer/Escribir</p>	<p>Definición: Es un identificador específico de un objeto.</p> <p>Uso: formas de identificar un objeto</p> <p>Formato: Cadena alfanumérica sin espacios internos.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Inicializado con una cadena vacía. El SCO es responsable de establecer este valor. Si el valor es pedido antes de ser establecido se debería devolver una cadena vacía. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.objectives.0.id") ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de establecimiento y devolución de datos: "A1317" "Obj123" <p>Ejemplos de uso del SCO: El SCO podría querer mostrar información resumida sobre los objeivos.</p> <pre>var objID = LMSGetValue("cmi.objectives.0.id") //Mostrar el ID objetivo para el SCO El SCO podría querer establecer el ID objetivo para razones informativas. LMSSetValue("cmi.objectives.0.id","Obj1")</pre>
<p>cmi.objectives.n.score Cada objetivo puede contener una puntuación asociada.</p> <p>Children de <code>cmi.objectives.n.score</code>:</p>	

raw, min, max	
cmi.objectives.n.score._children	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMISString255</p> <p>Accesibilidad del SCO: Sólo lectura</p>	<p>Definición: Las palabras clave de los children se usan para determinar todos los elementos dentro de la categoría cmi.objectives.n.score que son admitidos por el LMS. Si un elemento no tiene children pero se admite, se devuelve una cadena vacía. Si no se admite un elemento, no se devuelve nada pero una llamada a los errores verifica que el elemento no es admitido.</p> <p>Uso: Se usa para determinar qué elementos se admiten en el LMS.</p> <p>Formato: La respuesta es una lista separada por comas de todos los elementos admitidos por el LMS.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.objectives.0.score._children") ○ Ejemplo de valores devueltos: "raw,min,max" "raw" "" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 402 – Establecimiento de valores inválido, el elemento es una palabra clave ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO: El SCO podría usar este elemento para determinar qué elementos son admitidos por el LMS.</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives.0.score") if (objScoreChildren.indexOf("raw") != -1) { // Establecer la puntuación objetiva }</pre>
cmi.objectives.n.score.raw	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIDecimal or CMIBlank</p> <p>Accesibilidad SCO: Lectura/Escritura</p>	<p>Definición: Representación numérica del trabajo hecho por el alumno después de cada intento. No es obligatorio que sea procesado o tenido en cuenta.</p> <p>Debe ser un número entre 0 y 100</p> <p>Uso: Resultado bruto después de cada intento por un objetivo.</p> <p>Formato: Número decimal.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El elemento debe ser inicializado con una cadena vacía (""), El SCO es responsable de establecer ese valor. Si se pide el valor antes de que sea almacenado el LMS debe devolver una cadena vacía (""). ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.objectives.0.score.raw") ○ Código de error:

	<ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de establecimiento y devuelta de datos: "96.7" "5" "" <p>Ejemplo de uso del SCO: El SCO podría usar esto para establecer la puntuación bruta asociada con un objetivo.</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives.0.score") if (objScoreChildren.indexOf("raw") != -1) { LMSSetValue("cmi.objectives.0.score.raw", "85") }</pre>
cmi.objectives.n.score.max	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMIDecimal or CMIBlack</p> <p>Accesibilidad SCO: Lectura/escritura</p>	<p>Definición: La máxima puntuación o número total que el estudiante podría haber logrado sobre un objetivo.</p> <p>Debe ser un valor comprendido entre cero y 100.</p> <p>Uso: puntuación máxima después de cada intento para un objetivo.</p> <p>Formato: Número decimal o en blanco.</p> <p>Comportamiento LMS:</p> <ul style="list-style-type: none"> ▪ Inicilización: El elemento debería ser inicializado en una cadena vacía (""). El SCO es responsable de establecer este valor. Si se pide el valor del elemento antes de ser establecido, el LMS debería devolver una cadena vacía. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSGetValue("cmi.objectives.0.score.max") ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de establecimiento/devolución de valores: "100" "10" "" <p>Ejemplo de uso del SCO: El SCO podría usar esto para establecer una puntuación máxima a un objetivo.</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives.0.score") if (objScoreChildren.indexOf("max") != -1) { LMSSetValue("cmi.objectives.0.score.max", "5") }</pre>

}	
cmi.objectives.n.score.min	
<p>Llamadas al API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: Sí</p> <p>Tipo de dato: CMIDecimal or CMIBlank</p> <p>Accesibilidad SCO: Leer/Escribir</p>	<p>Definición: La mínima puntuación que el estudiante podría haber logrado con el objetivo. Debe estar comprendida entre cero y 100.</p> <p>Uso: Puntuación mínima después de cada realización de un objetivo.</p> <p>Formato: Número decimal o en blanco.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El elemento debe ser inicializado en una cadena vacía (""). El SCO es responsable de establecer ese valor. Si se pide el valor del elemento antes de que este sea establecido, entonces el LMS debería devolver una cadena vacía. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Example API call: LMSGetValue("cmi.objectives.0.score.min") ○ Error Code: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de establecimiento/devolución de datos: "0" "45.5" "" <p>Ejemplo de uso del SCO: El SCO podría usar esto para asociar una mínima puntuación con un objetivo.</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives.0.score") if (objScoreChildren.indexOf("min") != -1) { LMSSetValue("cmi.objectives.0.score.min", "2") }</pre>
cmi.objectives.n.status	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIVocabulary (Status)</p> <p>"passed" "completed" "failed" "incomplete" "browsed" "not attempted"</p> <p>Accesibilidad SCO: Lectura/Escritura</p>	<p>Definition: Es el estado obtenido por el estudiante en el objetivo del SCO. 6 valores posibles.</p> <p>Uso: Se usa para seguir la pista del estado de los estudiantes para un objetivo dado.</p> <p>Formato:</p> <ul style="list-style-type: none"> • passed: El número necesario de objetivos en el SCO han sido aprobados o se logró la puntuación para aprobar. • completed: El objetivo puede o no puede ser aprobado pero todos sus contenidos pueden haber sido vistos por el estudiante. Esto es lo que indica la variable completed. • failed: El estudiante ha suspendido los objetivos. En cambio esto no obliga a que todos los contenidos hayan sido vistos. • incomplete: El objetivo ha sido empezado pero no terminado • browsed: El estudiante ya ha lanzado el LMS antes cargando el objetivo. • not attempted: significa que el estudiante hizo un intento de cargar el curso pero por alguna razón el curso ni siquiera ha sido empezado. Quizás el alumno tan sólo ha leído el árbol de contenidos y decidió que no estaba preparado para afrontar el objetivo. Cualquier algoritmo del SCO puede ser usado para cambiar el valor "not attempted" a "incomplete"

	<p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Manejado por el SCO ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada alAPI: LMSGetValue("cmi.objectives.0.status") ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de establecimiento/devolución de valores: "not attempted" "passed" "browsed" <p>Ejemplo de uso del SCO: El SCO podría usar esto para establecer un estado asociado con un objetivo</p> <pre>var objScoreChildren = LMSGetValue("cmi.objectives._children") if (objScoreChildren.indexOf("status") != -1) { LMSSetValue("cmi.objectives.0.status","failed") }</pre>
<p>cmi.student_data Información para adaptar un SCO al perfil del estudiante.</p> <p>Children de cmi.student_data: mastery_score, max_time_allowed, time_limit_action</p>	
<p>cmi.student_data_children</p>	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMISString255</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Definición: La palabra clave _children se usa para determinar todos los elementos en la categoría principal que soporta el LMS. Si un elemento no tiene hijos (children) pero los soporta, se devuelve una cadena vacía. Si el elemento no es soportado por el LMS también se devuelve una cadena vacía pero podemos darnos cuenta haciendo un pedido sobre el último error en el cual se dirá que el elemento no es soportado por el LMS.</p> <p>Uso: Para determinar qué elementos de datos de cmi.student_data están admitidos por el LMS.</p> <p>Formato: El valor devuelto es una lista separada con comas del nombre de los elementos en la categoría principal que están soportados por el LMS</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> • Inicialización: El conjunto de _children soportados para este grupo. • LMSGetValue(): LMS devuelve una lista separada por comas de los elementos admitidos. <ul style="list-style-type: none"> ○ Llamada de ejemplo al API: LMSGetValue("cmi.student_data_children") ○ Ejemplo de valores devueltos: "mastery_score,time_limit_action,max_time_allowed" "max_time_allowed" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error de no implementación. Si el elemento no es soportado por el LMS se devolvería una cadena vacía • LMSSetValue(): El LMS debería establecer un código de error de acuerdo

	<p>con lo siguiente::</p> <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 402 – Valor establecido inválido: el elemento es una palabra clave. Si el elemento es admitido por el LMS y una llamada a LMSSetValue sobre este elemento debe colocar el código de error en 402. ▪ 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. <p>Ejemplo de uso del SCO: Los SCOs pueden usar llamadas al cmi.studen_data._children para determinar si un cierto elemento está implementado por el LMS.</p> <pre>var studentDataChildren = LMSGetValue("cmi.student_data._children"); if (studentDataChildren.indexOf("attempt_number") != -1) { var maxTimeAllowed = LMSGetValue("cmi.student_data.max_time_allowed"); }</pre>
cmi.student_data.mastery_score	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIDecimal</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Definición: La puntuación obtenida, determinada fuera del SCO. Cuando la puntuación del SCO es mayor o igual que la puntuación de referencia, se considera que el estudiante ha aprobado el SCO. Algunas veces el SCO no sabe cual es esa puntuación de referencia porque es determinada por el LMS.</p> <p>Uso: Para que un LMS admita mastery_score, se debe poder cambiar el elemento lesson_status basado en la puntuación obtenida en el SCO. Simplemente pasar un mastery_score a un SCO no quiere decir que se soporte completamente esta característica.</p> <p>Formato: Número decimal.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS es responsable – valor obtenido del manifiesto (adlcp:masteryscore) - ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de devolución de datos: "75" "100" "5" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. • LMSSetValue():El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 403 Los elementos son de sólo lectura . Si el elemento es admitido y se hace una llamada a LMSSetValue() en este elemento, entonces el LMS debería actualizar el código de error a 403. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO: var masteryScoreValue = LMSGetValue("cmi.student_data.mastery_score");</p>
cmi.student_data.max_time_allowed	
<p>Llamadas admitidas por el API: LMSGetValue()</p>	<p>Definición: La cantidad de tiempo que puede utilizar el estudiante en el intento actual de realizar el SCO. Mire time_limit_action para la respuesta esperada por el SCO cuando se excede el límite de tiempo disponible.</p>

<p>Obligatorio LMS: No</p> <p>Tipo de datos: CMITimespan</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Uso: Se usa para presentar ante el estudiante el máximo tiempo que hay disponible para realizar un SCO.</p> <p>Formato: horas, minutos y segundos separados por 2 puntos. HHHH:MM:SS.SS</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS es responsable – valor obtenido del manifiesto (adlcp:maxtimeallowed)- ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de valores devueltos: "00:14:30" "02:03:00" "01:09:00" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. • LMSSetValue():El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 403 Los elementos son de sólo lectura . Si el elemento es admitido y se hace una llamada a LMSSetValue() en este elemento, entonces el LMS debería actualizar el código de error a 403. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO: var maxTimeAllowedValue = LMSGetValue("cmi.student_data.max_time_allowed");</p>
cmi.student_data.time_limit_action	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIVocabulary (Time Limit Action) "exit,message" "exit,no message" "continue,message" "continue,no message"</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Definición: Dice al SCO que hacer cuando se excede el tiempo máximo.</p> <p>Formato: cuatro valores posibles:</p> <ul style="list-style-type: none"> • "exit,message" • "exit,no message" • "continue,message" • "continue,no message" <p>Comportamiento LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El LMS es responsable. El valor se obtiene del manifiesto (adlcp:timelimitaction) ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Ejemplo de valores devueltos: "exit,message""continue,no message" ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. • LMSSetValue():El LMS debería actualizar el código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 403 Los elementos son de sólo lectura . Si el elemento es admitido y se hace una llamada a LMSSetValue() en este elemento, entonces el LMS debería actualizar el código de error a 403. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido.

	<p>Ejemplo de uso del SCO:</p> <pre>var timeLimitActionValue = LMSGetValue("cmi.student_data.time_limit_action");</pre>
--	--

<p>cmi.student_preference Opciones seleccionadas que son apropiadas para siguientes SCOs.</p> <p>Children de cmi.student_preference: audio, language, speed, text</p>	
<p>cmi.student_preference.children</p>	
<p>Llamadas admitidas al API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMISring255</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Definición: La palabra clave <code>_children</code> se usa para determinar todos los elementos en la categoría principal que soporta el LMS. Si un elemento no tiene hijos (children) pero los soporta, se devuelve una cadena vacía. Si el elemento no es soportado por el LMS también se devuelve una cadena vacía pero podemos darnos cuenta haciendo un pedido sobre el último error en el cual se dirá que el elemento no es soportado por el LMS.</p> <p>Uso: Para determinar qué elementos de datos de <code>cmi.student_preference</code> están admitidos por el LMS.</p> <p>Formato: El valor devuelto es una lista separada con comas del nombre de los elementos en la categoría principal que están soportados por el LMS</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> • Inicialización: El conjunto de hijos soportados para este grupo. Por tanto una llamada a <code>LMSGetValue()</code> devolvería la lista apropiada de hijos soportados. • LMSGetValue(): LMS devuelve una lista separada por comas de los elementos admitidos. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error de no implementación. Si el elemento <code>cmi.core._children</code> no es soportado por el LMS se devolvería una cadena vacía. NOTA: el <code>cmi.core._children</code> debe ser admitido por el LMS ya que el elemento es obligatorio. • LMSSetValue(): El LMS debería establecer un código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 402 – Valor establecido inválido: el elemento es una palabra clave. Si el elemento es admitido por el LMS y una llamada a <code>LMSSetValue</code> sobre este elemento debe colocar el código de error en 402. ▪ 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. <p>Ejemplo de uso del SCO:</p> <pre>var studentPreferenceChildren = LMSGetValue("cmi.student_preference._children"); if (studentPreferenceChildren.indexOf("audio") != -1) { LMSSetValue("cmi.student_preference.audio", "10"); }</pre>
<p>cmi.student_preference.audio</p>	
<p>Llamadas admitidas por el API:</p>	<p>Definición: el audio se puede apagar y controlar su volumen. Este elemento indica si el audio está activado o desactivado.</p>

<p>LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMISInteger</p> <p>Accesibilidad SCO: Leer Escribir</p>	<p>Uso: Se usa por el SCO para mandar y obtener información del LMS sobre las preferencias de audio del estudiante.</p> <p>Formato: Dígitos desde el -1 hasta el 100. -1: apagado 0: el SCO usa por defecto su estado predefinido o el mismo estado de audio que el SCO anterior. 1 - 100: es el volumen, 100 quiere decir "tan alto como sea posible"</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Si lo admite, el LMS debería inicializar este valor en cero. Es responsabilidad del SCO regular este valor. Si se pide este valor antes de establecerse el LMS debe devolver una cadena vacía. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. ▪ LMSSetValue(): Situa el elemento del modelo de datos en el valor dado. Éste debe corresponder al tipo de datos para este elemento. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo e valores devueltos o almacenados: "-1" "0" "50" <p>Ejemplode uso del SCO: LMSSetValue("cmi.student_preference.audio", "10"); var audioValue = LMSGetValue("cmi.student_preference.audio");</p>
cmi.student_preference.language	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMISString255</p> <p>Accesibilidad SCO: Lectura Escritura</p>	<p>Definición: Para los SCOs con varios idiomas este elemento identifica en que idioma se deb presentar la información.</p> <p>Formato: cadena de letras, puede incluir espacios.</p> <p>Comportamiento LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: si lo admite el LMS, se debe inicializar con una cadena vacía. El SCO es el responsable de establecer este valor. Si se pide el valor de este elemento antes de ser establecido, el LMS debe devolver una cadena vacía. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. NOTA: El elemento debe ser soportado por el LMS ya que el elemento es obligatorio ▪ LMSSetValue(): Situa el elemento del modelo de datos en el valor dado. Éste debe corresponder al tipo de datos para este elemento. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es

	<p>admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido.</p> <ul style="list-style-type: none"> ▪ Ejemplo de valores devueltos/almacenados "English" "French" <p>Ejemplo de uso del SCO: LMSSetValue("cmi.student_preference.language", "English"); var languageValue = LMSGetValue("cmi.student_preference.language");</p>
cmi.student_preference.speed	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMISInteger</p> <p>Accesibilidad SCO: Leer Escribir</p>	<p>Definición: Los SCOs algunas veces pueden ser difíciles de asimilar porque van o muy rápidos o muy lentos. Este elemento controla la velocidad de entrega de contenido.</p> <p>Uso: Se usa por el SCO para establecer y obtener del LMS las preferencias sobre velocidad del alumno.</p> <p>Formato: Dígito desde -100 hasta 100. -100 es la velocidad menor 0: el SCO usa la velocidad que tiene por defecto 100 es la máxima velocidad disponible en el sistema</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Si el elemento se admite en el LMS, este elemento debe ser inicializado en cero. Es responsabilidad del SCO el establecimiento de este valor. Si se pide un valor de este elemento antes de que sea establecido se debe devolver una cadena vacía. ▪ LMSGetValue(): Returns the current value stored by the LMS. <ul style="list-style-type: none"> ○ Código de error: 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. NOTA: El elemento debe ser soportado por el LMS ya que el elemento es obligatorio ▪ LMSSetValue(): Situa el elemento del modelo de datos en el valor dado. Éste debe corresponder al tipo de datos para este elemento. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de almacenamiento/devolución de valores: "-100" "0" "3" <p>Ejemplo de uso del SCO: LMSSetValue("cmi.student_preference.speed", "-100"); var speedValue = LMSGetValue("cmi.student_preference.speed");</p>
cmi.student_preference.text	
<p>Llamadas admitidas por el API: LMSGetValue() LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMISInteger</p>	<p>Definición: En un SCO con contenidos en audio se puede preferir ver el contenido de audio en una ventana de texto en lugar de escucharlo. Este elemento identifica si el texto del audio aparece en la pantalla.</p> <p>Formato: Uno de estos tres dígitos: -1 : no se muestra el texto 0 : valor por defecto en el SCO 1 : el texto se muestra.</p>

<p>Accesibilidad SCO: Leer Escribir</p>	<p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: Si es admitida por el LMS este elemento se debería inicializar con "0". Es responsabilidad del SCO establecer este valor. Si el valor es pedido antes de ser establecido entonces el LMS debe devolver una cadena vacía. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: 401 – Error de no implementación. Si el elemento no está soportado el código de error se establece en 401 por el LMS para indicar que el elemento no es soportado. NOTA: El elemento debe ser soportado por el LMS ya que el elemento es obligatorio ▪ LMSSetValue(): Situa el elemento del modelo de datos en el valor dado. Éste debe corresponder al tipo de datos para este elemento. <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 405 – Tipo de dato incorrecto. Si el elemento es admitido (el elemento debe ser admitido por el LMS dado que es obligatorio) y una llamada invoca a LMSSetValue() con un valor que no es el tipo de dato correcto. ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ Ejemplo de almacenamiento/devolución: "0" "-1" "1" <p>Ejemplo de uso en el SCO: LMSSetValue("cmi.student_preference.text", "-1"); var textValue = LMSGetValue("cmi.student_preference.text");</p>
--	--

cmi.interactions

En este contexto, una interacción es una entrada o grupo de entradas de datos reconocidos y regrabables desde el estudiante hasta el ordenador.

Todos los elementos de este grupo están relacionados con una entrada de datos reconocida y regrabable.. Esta categoría recopila información detallada de cada interacción medida mientras el estudiante trabaja con el SCO.

Children de cmi.interactions:

id, objectives, time, type, correct_responses, weighting, student_response, result, latency

cmi.interactions._children

<p>Llamadas admitidas por el API: LMSGetValue() Obligatorio LMS: No Tipo de dato: CMIStrng255 Accesibilidad SCO: Sólo lectura</p>	<p>Idem al uso de children en las demás categorías</p>
--	--

cmi.interactions._count

<p>Llamadas admitidas por el API: LMSGetValue() Obligatorio LMS: No Tipo de dato: CMIStrng255 Accesibilidad SCO: Sólo lectura</p>	<p>Idem que otros usos de la palabra clave _count</p>
--	---

cmi.interactions.n.id	
<p>Llamadas admitidas por el API: LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIIentifier</p> <p>Accesibilidad SCO: Sólo escritura</p>	<p>Definición: Un único identificador por cada interacción</p> <p>Uso: Se usa para establecer un único identificador de interacción con cada SCO.</p> <p>Formato: Cadena alfanumérica, sin espacios.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: el valor es controlado por el SCO. ▪ LMSGetValue(): El LMS debería establecer un código de error de acuerdo con lo siguiente: <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ 404 – El elemento es sólo escritura. Si el SCO intenta ejecutar LMSGetValue() con este elemento. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.interactions.0.id","I_001") ○ Ejemplo de establecimiento de valores: "I_001" "i1" ○ Código de error: <ul style="list-style-type: none"> ▪ 205 – Tipo de dato incorrecto: ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO: // Establecer los primeros identificadores de interacción. LMSSetValue("cmi.interactions.0.id","I_001");</p>
cmi.interactions.n.objectives	
cmi.interactions.n.objectives_count	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIIInteger</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Idem que otros usos de la palabra clave _count</p> <p>Ejemplo de uso del SCO: El SCO podría usar el elemento _count para determinar que índice de array usar más adelante en el proceso.</p> <pre>// conseguir la cuenta de identificadores objetivos grabados por el LMS en una interacción dada var totalIDs = LMSGetValue("cmi.interactions.0.objectives_count") // El valor devuelto desde el LMS es el número total de registros var request = "cmi.interactions.0.objectives." + totalID + ".id" // Establecer las interacciones identificador objetivo LMSSetValue(request, "ObjID-110")</pre>
cmi.interactions.n.objectives.n.id	
<p>Llamadas admitidas por el API: LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMIIentifier</p> <p>Accesibilidad SCO:</p>	<p>Definición: identificador de objetivos creado por el autor.</p> <p>Uso: Se usa para identificar el objetivo para el que es la interacción.</p> <p>Formato: Cadena alfanumérica, sin espacios.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: el valor es controlado por el SCO. ▪ LMSGetValue():

<p>Sólo escritura</p>	<ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ 404 – El elemento es sólo escritura. Si el SCO intenta ejecutar LMSGetValue() con este elemento. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.interactions.0.objectives.0.id","A1333") ○ Ejemplo de establecimiento de valores: "A1333" "Obj123" ○ Código de error: <ul style="list-style-type: none"> ▪ 205 – Tipo de dato incorrecto ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO: // establecer el ID objetivo para la interacción dada LMSSetValue("cmi.interactions.0.objectives.0.id","A1333")</p>
cmi.interactions.n.time	
<p>Llamadas admitidas por el API: LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMITime</p> <p>Accesibilidad SCO: Sólo escritura</p>	<p>Definición: Identificación de cuándo la comunicación del estudiante se finalizó.</p> <p>Uso: Se usa como un indicativo del tiempo para la comunicación.</p> <p>Formato: Un momento dado identificado en horas, minutos y segundos dentro de un reloj de 24 horas.</p> <p>Comportamiento LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El valor es controlado por el SCO. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. ▪ 404 – El elemento es sólo escritura. Si el SCO intenta ejecutar LMSGetValue() con este elemento. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.interactions.0.time","12:33:35.5") ○ Ejemplo de estable: "12:33:35.5" "22:30:40" ○ Código de error: <ul style="list-style-type: none"> ▪ 205 – Tipo de dato incorrecto ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO: // Establecer el tiempo de comunicación LMSSetValue("cmi.interactions.0.time","12:33:35.5")</p>
cmi.interactions.n.type	
<p>Llamadas soportadas por el API: LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIVocabulary</p>	<p>Definición: Indicación de qué categoría de comunicación está registrada. El tipo de comunicación interpreta cómo la comunicación de respuesta debe ser interpretada.</p> <p>Uso: Para indicar el tipo de comunicación que se está llevando a cabo.</p> <p>Formato:</p> <ul style="list-style-type: none"> • "true-false": una pregunta con sólo 2 respuestas posibles

<p>(Interaction) "true-false" "choice" "fill-in" "matching" "performance" "sequencing" "likert" "numeric"</p> <p>Accesibilidad SCO: Sólo escritura</p>	<ul style="list-style-type: none"> ● "choice": Una pregunta con un número limitado de respuestas predefinidas. El alumno debe elegir una o varias respuestas. ● "fill-in": una pregunta con una respuesta simple de una o varias palabras. La respuesta no está predefinida, debe introducirla el estudiante. ● "matching": Una pregunta con 2 conjuntos de objetos para relacionar entre sí. ● "performance": Una cuestión "performance" es, en algunos sentidos, similar a las preguntas con múltiples respuestas para escoger. Sin embargo, en lugar de seleccionar una respuesta escrita, el estudiante debe realizar una tarea o acción. ● "sequencing": En una cuestión "sequencing", el estudiante tiene que ordenar una lista de valores. ● "likert": Una cuestión "likert" ofrece al estudiante un grupo de alternativas en una secuencia. La respuesta generalmente está basada en la opinión de los estudiante. ● "numeric": la respuesta es un número simple con o sin decimales. Debe estar comprendida en un intervalo dado para que sea correcta. <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ■ Inicialización: El valor es controlado por el SCO. ■ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ■ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ■ 404 – El elemento es de solo escritura. Si un SCO intenta llamar a LMSGetValue() en este elemento, el LMS debe establecer el código de error en 404 y devolver una cadena vacía. ■ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.interactions.0.type","likert") ○ Ejemplo de establecimiento de valores: "likert" "true-false" "performance" ○ Error Code: <ul style="list-style-type: none"> ■ 205 –Tipo de dato incorrecto ■ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. <p>Ejemplo del uso SCO: // establecer el tipo de comunicación LMSSetValue("cmi.interactions.0.type","choice");</p>
cmi.interactions.n.correct_responses	
cmi.interactions.n.correct_responses._count	
<p>Llamadas admitidas por el API: LMSGetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de datos: CMIIInteger</p> <p>Accesibilidad SCO: Sólo lectura</p>	<p>Idem que uso de _count en otros elementos</p> <p>Ejemplo del uso del SCO: El SCO podría usar este elemento para determinar qué índice array se podría usar más tarde en el proceso.</p> <p>// obtener la cuenta de identificadores objetivos registrados por el LMS en una comunicación dada. var totalPattern = LMSGetValue("cmi.interactions.0.correct_responses._count")</p> <p>// El valor devuelto por el LMS es el número total de registros</p> <p>var request = "cmi.interactions.0.correct_responses." + totalPattern + ".pattern"</p> <p>// Establecer el formato de respuestas correctas para la comunicación. LMSSetValue(request, "t")</p>
cmi.interactions.n.correct_responses.n.pattern	

<p>Llamadas admitidas por el API: LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIFeedback</p> <p>Accesibilidad SCO: Sólo escritura</p>	<p>Definición: Descripción de las posibles respuestas del estudiante a ala comunicación. Puede haber más de una respuesta correcta y algunas respuestas pueden ser más correctas que otras.</p> <p>Uso: Este es el modelo con las respuestas correctas al ejercicio entero suministrado por el SCO.</p> <p>Formato: Si el cmi.interactions.n.type es: true-false: Entonces el modelo de respuestas es un sólo carácter o número. Los caracteres admitidos son:0, 1, t y f. Cero corresponde a falso. Si la respuesta es una palabra entera (por ejemplo "true"), sólo el primer carácter es tenido en cuenta. choice: Entonces el modelo de respuesta es uno o más caracteres separados por una coma. Se pueden usar enteros (0-9), letras (a-z) o ambos. Cada posible respuesta está limitada a un sólo carácter. Si hay más de 26 posibilidades se debe usar el tipo "performance". fill-in: Entonces el modelo de respuesta es una cadena alfanumérica. Los espacios sólo se pueden utilizar después del primer carácter imprimible. numeric: Entonces el modelo de respuesta es un solo número. El número puede o no puede tener decimales. likert: No hay respuesta incorrecta para una pregunta likert. El campo se puede dejar en blanco. matching: Entonces el modelo de respuesta son parejas de identificadores separados por un punto. performance: Entonces el modelo de respuesta es un campo alfanumérico limitado a 255 caracteres. sequencing: Entonces el modelo de respuesta es una lista de elementos en cualquier orden. La posición final de los elementos se usa para determinar si son correctos, no para indicar el orden.</p> <p>Comportamiento LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: el valor es controlado por el SCO ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ 404 – El elemento es de solo escritura. Si un SCO intenta llamar a LMSGetValue() en este elemento, el LMS debe establecer el código de error en 404 y devolver una cadena vacía. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: Si cmi.interactions.n.type está activa entonces: LMSSetValue("cmi.interactions.0.correct_responses.0.pattern","1.c,2.b,3.a,4.d") ○ Ejemplo de almacenamiento de valores: "1.c,2.b,3.a,4.d" "t" ○ Código de error: <ul style="list-style-type: none"> ▪ 205 –Tipo de dato incorrecto ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. <p>Ejemplo de uso del SCO: // establecer el modelo de respuesta // el tipo de cuestión es true-false LMSSetValue("cmi.interactions.0.correct_responses.0.pattern","t");</p>
cmi.interactions.n.weighting	
<p>Llamadas admitidas por el API: LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato:</p>	<p>Definición: Los ejercicios varían en importancia. Este elemento se usa para diferenciar la diferente importancia de los ejercicios. Por ejemplo, si el primer ejercicio tiene un peso de 15 y el segundo tiene un peso de 25, la puntuación total estará más influida por el segundo ejercicio. Un peso de cero quiere decir que el ejercicio no se tiene en cuenta para la nota final. También se puede dar pesos diferentes a preguntas distintas dentro de un ejercicio complejo.</p>

<p>CMIDecimal</p> <p>Accesibilidad SCO: Sólo escritura</p>	<p>Formato: número decimal (o entero)</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El valor es controlado por el SCO. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ 404 – El elemento es de solo escritura. Si un SCO intenta llamar a LMSGetValue() en este elemento, el LMS debe establecer el código de error en 404 y devolver una cadena vacía. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamadas al API: LMSSetValue("cmi.interactions.0.weighting","0.66") ○ Ejemplo de establecimiento de valores: "0.66" "0" ○ Código de error: <ul style="list-style-type: none"> ▪ 205 –Tipo de dato incorrecto ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. <p>Ejemplo de uso del SCO: // Establecer el peso para el ejercicio LMSSetValue("cmi.interactions.0.weighting","0");</p>
cmi.interactions.n.student_response	
<p>Llamadas admitidas al API LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIFeedback</p> <p>Accesibilidad SCO: Sólo escritura</p>	<p>Definición: Descripción de posibles respuestas al ejercicio. Puede haber más de una respuesta correcta y ser alguna más correcta que las demás.</p> <p>Uso: Esta es la respuesta que ha dado el estudiante al ejercicio. Este valor puede ser comparado con cmi.interactions.n.correct_responses.n.pattern.</p> <p>Formato: Si cmi.interactions.n.type es: true-false: Entonces el modelo de respuestas es un sólo carácter o número. Los caracteres admitidos son:0, 1, t y f. Cero corresponde a falso. Si la respuesta es una palabra entera (por ejemplo "true"), sólo el primer carácter es tenido en cuenta. choice: Entonces el modelo de respuesta es uno o más caracteres separados por una coma. Se pueden usar enteros (0-9), letras (a-z) o ambos. Cada posible respuesta está limitada a un sólo carácter. Si hay más de 26 posibilidades se debe usar el tipo "performance". fill-in: Entonces el modelo de respuesta es una cadena alfanumérica. Los espacios sólo se pueden utilizar después del primer carácter imprimible. numeric: Entonces el modelo de respuesta es un solo número. El número puede o no puede tener decimales. likert: No hay respuesta incorrecta para una pregunta likert. El campo se puede dejar en blanco. matching: Entonces el modelo de respuesta son parejas de identificadores separados por un punto. performance: Entonces el modelo de respuesta es un campo alfanumérico limitado a 255 caracteres. sequencing: Entonces el modelo de respuesta es una lista de elementos en cualquier orden. La posición final de los elementos se usa para determinar si son correctos, no para indicar el orden.</p> <p>Comportamiento LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El valor es controlado por el SCO. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido.

	<ul style="list-style-type: none"> ▪ 404 – El elemento es de solo escritura. Si un SCO intenta llamar a LMSGetValue() en este elemento, el LMS debe establecer el código de error en 404 y devolver una cadena vacía. <ul style="list-style-type: none"> ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: Si cmi.interactions.n.type está encajando con lo correcto, entonces LMSSetValue("cmi.interactions.0.student_response","1.c,2.b,3.a,4.d") ○ Ejemplo de establecimiento de valores: "t" "1.c,2.b,3.c,4.d" ○ Código de error: <ul style="list-style-type: none"> ▪ 205 –Tipo de dato incorrecto ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. <p>Ejemplo de uso del SCO: // establecer la respuesta del estudiante // La respuesta del estudiante fue verdadero-falso LMSSetValue("cmi.interactions.0.student_response","f");</p>
cmi.interactions.n.result	
<p>Llamadas admitidas por el API: LMSSetValue()</p> <p>Obligatorio LMS: No</p> <p>Tipo de dato: CMIVocabulary (Result) "correct" "wrong" "unanticipated" "neutral" "x.x" (CMIDecimal)</p> <p>Accesibilidad SCO: Sólo escritura</p>	<p>Definición: como el sistema supervisa la respuesta del estudiante.</p> <p>Uso: Esta es el resultado a la respuesta del estudiante.</p> <p>Formato:</p> <ul style="list-style-type: none"> • "correct" • "wrong" • "unanticipated" • "neutral" • "x.x" (CMIDecimal) <p>Comportamiento LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: El valor es controlado por el SCO. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Error Code: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ 404 – El elemento es de solo escritura. Si un SCO intenta llamar a LMSGetValue() en este elemento, el LMS debe establecer el código de error en 404 y devolver una cadena vacía. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de llamada al API: LMSSetValue("cmi.interactions.0.result","correct") ○ Ejemplo de valores establecidos: "correct" "95.5" "unanticipated" ○ Error Code: <ul style="list-style-type: none"> ▪ 205 – Tipo de dato incorrecto ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía. Y el código de error se actualiza indicando que el elemento no es admitido. <p>Ejemplo de uso del SCO: // establecer el resultado para la pregunta LMSSetValue("cmi.interactions.0.result","correct");</p>
cmi.interactions.n.latency	
<p>Llamadas admitidas por el API:</p>	<p>Definición: tiempo desde que se inició el ejercicio puntuable.</p>

<p>LMSSetValue() Obligatorio LMS: No Tipo de dato: CMITimespan Accesibilidad SCO: Sólo escritura</p>	<p>Format: Horas, minutos y segundos separados por dos puntos. HHHH:MM:SS.SS Horas tiene un mínimo de 2 dígitos y un máximo de 4 dígitos. Los minutos consistirán en 2 dígitos exactos. Los segundos tendrán 2 dígitos con un punto decimal opcional para indicar décimas o centésimas.</p> <p>Comportamiento del LMS:</p> <ul style="list-style-type: none"> ▪ Inicialización: el valor es controlado por el SCO. ▪ LMSGetValue(): <ul style="list-style-type: none"> ○ Código de error: <ul style="list-style-type: none"> ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. ▪ 404 – El elemento es de solo escritura. Si un SCO intenta llamar a LMSGetValue() en este elemento, el LMS debe establecer el código de error en 404 y devolver una cadena vacía. ▪ LMSSetValue(): <ul style="list-style-type: none"> ○ Ejemplo de devolución de datos: LMSSetValue("cmi.interactions.0.latency","00:29:00") ○ Ejemplo de establecimiento de valores: "00:29:00" "1234:44:30" ○ Código de error: <ul style="list-style-type: none"> ▪ 205 – Tipo de dato incorrecto ▪ 401 – Error no implementado. Si este elemento no es admitido se devuelve una cadena vacía y un código de error se establece para indicar que el elemento no es admitido. <p>Ejemplo de uso del SCO: // establecer tiempo de ejercicio LMSSetValue("cmi.interactions.0.latency","00:20:00");</p>
--	---

1.10.7 Tipos de dato y vocabulario limitado

Existe una definición de tipo de dato por cada elemento en el modelo de datos de AICC CMI. A continuación se definen como el API y el modelo de datos debe ser implementado respecto al modelo de datos.

Tipo de dato	Descripción
CMIBlank	Cadena vacía ("").
CMIBoolean	"true" o "false".
CMIDecimal	Un número que puede tener un punto decimal. Si no le precede un signo menos, se entiende que el número es positivo.

<p>CMIFeedback</p>	<p>Una descripción estructurada de la respuesta de un estudiante a un ejercicio. La estructura y contenido depende del tipo de ejercicio. Los ejercicios definidos actualmente son:</p> <p>True-false: Feedback es uno de los siguientes caracteres: "0", "1", "t" o "f"</p> <p>Choice: feedback es una lista de uno o más caracteres separados por comas. Sólo son válidos los caracteres a-z y 0-9. Si todas las opciones deben ser seleccionadas para que el ejercicio sea correcto, entonces la lista debe estar comprendida en corchetes { }</p> <p>Fill in: cualquier cadena alfanumérica hasta 255 caracteres de longitud. Después de la primera letra los espacios se tienen en cuenta.</p> <p>Numeric: CMIDecimal</p> <p>Likert: un único carácter. Sólo válidos los caracteres a-z y 0-9</p> <p>Matching: uno o más pares de identificadores. Cada identificador es una sólo letra o número (0 -9 y a-z). Los identificadores en los pares están separados por un punto. Los pares están separados mediante otros pares mediante comas. Si todos los pares deben ser correctos para considerar el ejercicio aprobado entonces van comprendidos en corchetes { }</p> <p>Perfomance: formato muy flexible. Esencialmente una cadena alfanumérica de 255 caracteres o menos.</p> <p>Sequencing: una serie de caracteres separados por comas. Sólo validos 0-9 y a-z. El orden de los caracteres determina determina si el ejercicio es correcto.</p>
<p>CMIIentifier</p>	<p>Una cadena alfanumérica sin espacios en blanco hasta 255 caracteres</p>
<p>CMIInteger</p>	<p>Un entero de 0 a 65536.</p>
<p>CMISInteger</p>	<p>Un entero con signo desde -32768 a +32768</p>
<p>CMIStrng255</p>	<p>Un conjunto de caracteres ASCII hasta 255 caracteres.</p>
<p>CMIStrng4096</p>	<p>Un conjunto de caracteres ASCII con una máxima longitud de 4096 caracteres.</p>
<p>CMITime</p>	<p>Un momento dado en un reloj de 24 horas en la forma HH:MM:SS.SS. Las décimas y centésimas son opcionales</p>
<p>CMITimespan</p>	<p>Un tiempo dado de la forma HHHH:MM:SS.SS. Las horas deben tener un mínimo de 2 dígitos y un máximo de 4. El resto es obligatorio excepto las décimas y centésimas que son opcionales.</p>
<p>CMIVocabulary</p>	<p>Se usa para adjuntar vocabulario específico</p>

