

Facultad de Informática UNLP



Una aplicación móvil que integra las experiencias educativas de los alumnos de la Facultad de Informática

Carrera: Licenciatura en Sistemas

Autores:

Rossi Benjamín, 5813/7

Vazzano Andrés, 5814/8

Directores:

Queiruga Claudia

Díaz Javier

| | |
|--|-----------|
| Motivación | 5 |
| Introducción | 5 |
| Capítulo 1 | 8 |
| 1.1 Objetivos | 8 |
| 1.2 Caso de estudio: Sistemas de aprendizaje de la Facultad de Informática | 8 |
| 1.2.1 Caracterización de la problemática | 8 |
| Capítulo 2 | 10 |
| 2.1 Tecnologías de la información aplicadas a la educación | 10 |
| 2.1.1 Origen | 10 |
| 2.1.2 Definición de e-learning | 11 |
| 2.1.3 Ventajas y desventajas del e-Learning | 12 |
| 2.1.4 Uso de TIC en educación | 14 |
| 2.2 Estándares | 14 |
| 2.2.1 Definición de estándares orientados al e-learning | 14 |
| 2.2.2 Ventajas aportadas por los estándares en e-learning | 15 |
| 2.2.3 Organismos participantes en la estandarización en e-learning | 16 |
| 2.2.4 Estandarización: aspectos y procesos | 17 |
| 2.2.5 Objetos de aprendizaje | 18 |
| 2.2.6 Limitaciones de los estándares | 18 |
| 2.2.7 Especificaciones y estándares con mayor aceptación | 18 |
| 2.3 LMS: Learning management system | 19 |
| 2.3.1 Definición | 19 |
| 2.3.2 Evolución | 19 |
| 2.3.3 Propósito | 19 |
| 2.3.4 Ventajas | 20 |
| 2.3.5 Desventajas | 20 |
| 2.3.5.1 Problema de trabajo colaborativo entre los alumnos | 21 |
| 2.3.5.2 Problema de los LMS para ser accedidos desde dispositivos móviles | 21 |
| 2.3.5.3 Dificultad de los LMS para interactuar con otros sistemas | 21 |
| Capítulo 3 | 23 |
| 3.1 SCORM: Sharable Content Object Reference Model | 23 |
| 3.1.1 La especificación de SCORM | 23 |
| 3.1.2 La organización de SCORM | 23 |
| 3.1.3 Modelo de agregación de contenidos en SCORM | 24 |
| 3.1.4 Entorno de Tiempo de Ejecución | 24 |
| 3.1.5 Secuenciamiento y Navegación | 25 |
| 3.1.6 Ventajas de utilizar SCORM | 26 |
| 3.1.7 Inconvenientes de utilizar SCORM | 26 |

| | |
|--|-----------|
| 3.2 Tin Can | 27 |
| 3.2.1 Origen | 27 |
| 3.2.2 Característica que agrega a SCORM | 28 |
| 3.2.3 ¿Cómo funciona Tin Can API? | 28 |
| 3.2.4 ¿Cómo se estructura Tin Can? | 29 |
| 3.2.5 Núcleo de Tin Can | 30 |
| 3.2.6 Capacidades de Tin Can | 31 |
| 3.2.6.1 Simplicidad | 31 |
| 3.2.6.2 Independiente del navegador | 31 |
| 3.2.6.3 Distribución de Contenido | 31 |
| 3.2.6.4 Acceso a la información | 32 |
| 3.2.6.5 Variables definidas por el usuario | 33 |
| 3.2.6.6 Aprendizaje no inicializado en el LMS | 33 |
| 3.2.6.7 Juegos y Simulaciones | 33 |
| 3.2.6.8 Seguimiento de actividades en el mundo real | 34 |
| 3.2.6.9 Seguimiento Desconectado | 35 |
| 3.2.6.10 Cambios de plataforma | 35 |
| 3.2.6.11 Interoperabilidad | 35 |
| 3.2.6.12 Secuenciación | 36 |
| 3.2.6.13 Interacción entre alumnos e instructores durante el entrenamiento | 36 |
| 3.2.6.14 Aprendizaje colaborativo y aprendizaje basado en el equipo | 37 |
| 3.2.6.15 Etiquetar contenido con capacidades | 37 |
| 3.2.7 Debilidades de Tin Can | 38 |
| 3.2.7.1 Acuerdo de definición de actividad | 38 |
| 3.2.7.2 Variación de verbos | 38 |
| 3.2.7.3 Complejidad de informes | 38 |
| 3.3 LRS: Learning record store | 39 |
| 3.3.1 Funcionamiento del LRS | 40 |
| 3.3.1.1 Registrando experiencias educativas | 40 |
| 3.3.1.2 Actividades en el LRS | 40 |
| 3.3.1.3 Acceso de usuarios | 41 |
| 3.3.2 APIs especificadas por el estándar Tin Can | 42 |
| 3.3.3 Implementaciones de LRS | 43 |
| 3.3.3.1 LRS stand alone | 43 |
| 3.3.3.2 Integrando el LRS con el LMS | 44 |
| 3.3.4. Almacenamiento y gestión de datos | 45 |
| Capítulo 4 | 46 |
| 4.1 Especificación del prototipo | 46 |
| 4.2 Prototipo del LRS | 47 |
| 4.2.1 Requerimientos | 47 |
| 4.2.2 Casos de uso | 48 |

| | |
|---|-----------|
| 4.2.2.1 Diagrama de casos uso del LRS | 49 |
| 4.2.2.2 Descripción de los casos de uso del LRS | 49 |
| 4.2.3 Modelo | 53 |
| 4.2.4 Descripción de las clases principales del modelo | 54 |
| 4.2.5 LRS interfaz pública | 55 |
| 4.3 Extensión para Moodle | 56 |
| 4.4 Aplicación móvil | 56 |
| 4.3.1 Alcance | 56 |
| 4.3.2 Requerimientos | 57 |
| Capítulo 5 | 58 |
| 5.1 El LRS | 58 |
| 5.1.1 Herramientas para el desarrollo | 58 |
| 5.1.2 Implementación del modelo | 59 |
| 5.1.3 Implementación de la capa de negocio | 60 |
| 5.1.4 Statement API | 60 |
| 5.1.4.1 Implementando la Statement API | 61 |
| 5.2 Desarrollo del plugin para moodle | 62 |
| 5.2.1 Extensiones locales personalizadas | 62 |
| 5.2.2 Características de las extensiones | 62 |
| 5.2.3 Configuración de la extensión | 65 |
| 5.3 Desarrollo de la aplicación móvil | 69 |
| 5.3.1 Aplicaciones híbridas | 69 |
| 5.3.2 Estructura del LRSapp | 71 |
| Capítulo 6 | 74 |
| 6.1 Experiencias de usuarios | 74 |
| 6.1.1 Interfaz | 74 |
| 6.1.2 Menú principal | 75 |
| 6.1.3 Configuración inicial | 75 |
| 6.1.4 Cuentas | 76 |
| 6.1.5 Listar Statements | 79 |
| 6.1.6 Buscar Statements | 81 |
| 6.1.7 Crear Statement | 83 |
| Capítulo 7 | 84 |
| 7.1 Trabajos Futuros | 84 |
| 7.1.1 Completar la funcionalidad provista por el estándar Tin Can | 84 |
| 7.1.2 Implementar single sign on | 84 |
| 7.1.3 Implementación de estadísticas | 84 |
| 7.1.4 Mejoramiento de la interfaz de usuario del LRSapp | 85 |
| 7.1.5 Pruebas del campo | 85 |
| 7.2 Conclusión | 85 |

Motivación

Hoy en día vivimos en un mundo hiperconectado, donde el acceso a la información se produce en ámbitos que antes eran impensados; un micro, una sala de espera e inclusive en el mismo buffet de la facultad.

He de aquí, que, como alumnos que somos de la Facultad de Informática, hemos notado que el acceso a los distintos sistemas de la misma unidad académica se vuelve engorroso desde un dispositivo móvil. Son sistemas que no fueron pensados para ser accedidos desde un dispositivo móvil. A esto se suma que, generalmente, se requiere consultar más de un sistema y los mismos no se encuentran interconectados entre sí, de manera tal que el alumno pueda verlos a todos ellos como uno solo.

Para poder entender esta problemática veamos el siguiente ejemplo de la vida cotidiana de un alumno de nuestra facultad:

Es lo habitual que un alumno curse varias materias en un semestre. Las materias por él cursadas no necesariamente utilizan la misma plataformas de apoyo a los alumnos. Ejemplo, asignaturas como “Matemática 4” o “Sistemas y Organizaciones” utilizan el Moodle como nexos entre el alumno y el profesor. Sin embargo, otras materias, como en el caso de “Ingeniería de Software” utilizan la plataforma WebUNLP. Esta pluralidad de plataformas obliga al alumno a tener que visitar periódicamente ambas plataformas para mantenerse al corriente con las asignaturas.

Esto no sólo es engorroso, sino que el alumno se ve desmotivado a aprovechar los recursos que la facultad pone a su disposición. Informarse de fechas de exámenes, realizar consultas, enterarse de la entrega de nuevos contenidos, teniendo que chequear periódicamente distintas plataformas hace que el alumno no les saque el máximo provecho. Si para acceder a las distintas materias tiene que crear distintas cuentas de usuario, verificarlas periódicamente, llega un punto en que directamente ya no utiliza la plataforma.

La problemática planteada nos motivó a pensar en diseñar una herramienta que permita a los estudiantes acceder en forma unificada a estos sistemas de aprendizaje de la Facultad de Informática.

Introducción

El auge del e-learning[1][2] ha propiciado la proliferación de una gran variedad de sistemas orientados al aprendizaje, denominados LMS (Learning Management System)[3][4]. La mayoría de ellos son sistemas web que han contribuido enormemente a los procesos de formación [5][6].

Hoy en día, los alumnos de la Facultad de Informática, cuentan con diversas herramientas de este tipo, a saber:

- Sitio Web de la Facultad de Informática (www.info.unlp.edu.ar)
- Sistema de Administración de Cursos basado en Moodle [7] (catedras.info.unlp.edu.ar).
- Sistema Integrado de Gestión de Bibliotecas - Meran [8] (catalogo.info.unlp.edu.ar).

A pesar de los grandes aportes que estos sistemas brindan al proceso educativo, los mismos presentan ciertas desventajas para los tiempos que corren:

- No fueron pensados para ser accedidos desde dispositivos móviles, por lo tanto el acceso mediante este tipo de dispositivos es poco amigable.
- Los distintos sistemas no les representan a los alumnos una sola entidad de aprendizaje, por el contrario, son percibidos como entidades separadas, aunque estos LMS compartan contenido y finalidad .
- Las interfaces de los LMS suelen ser muy complejas, las mismas no se encuentran adaptadas para ser visualizadas desde dispositivos como celulares o tablets, lo cual resulta engorroso el acceso a las mismas mediante dispositivos móviles.

La tesina que aquí presentamos propone integrar los diferentes LMS y poner disponibles los recursos contenidos en ellos desde un dispositivo móvil. Hemos organizado el informe de la siguiente manera:

En el capítulo 1 describimos los objetivos generales y específicos de la tesina. A su vez planteamos el caso de estudio para poder entender la problemática en cuestión.

En el capítulo 2 introducimos el concepto de e-learning, evaluamos sus ventajas y desventajas y su uso en las TIC. Luego describimos los estándares que regulan el e-learning para finalizar el capítulo describiendo la componente de software LMS.

En el capítulo 3 describimos el estándar SCORM[9][10], analizando sus ventajas y desventajas y como él mismo ha evolucionado en el estándar Tin Can[11][12]. Para finalizar el capítulo analizamos qué es un LRS[13] [14] y sus distintas implementaciones.

En el capítulo 4 describimos la especificación del prototipo siguiendo la especificación del estándar Tin Can. Definimos la arquitectura y las distintas componentes que tendrá el desarrollo.

En el capítulo 5 describimos las principales características de cada unas de las componentes de la solución desarrollada: LRS, aplicación móvil y a extensión para Moodle. A su vez explicamos algunas de las decisiones de diseño que se tomaron a lo largo del desarrollo.

En el capítulo 6 describimos el funcionamiento del LRS a través de su aplicación

para móviles, el LRSapp. Mediante este prototipo, los alumnos accederán a sus experiencias educativas y generarán nuevas experiencias.

En el capítulo 7 listamos las tareas pendientes para trabajos futuros y presentamos nuestras conclusiones sobre la tesina.

Capítulo 1

1.1 Objetivos

Objetivo General

El objetivo de nuestra tesina de grado es enriquecer la experiencia de aprendizaje de los estudiantes de la Facultad de Informática integrando las diferentes herramientas educativas que ofrece la Facultad.

Objetivos Específicos

- Estudiar el estándar SCORM y su evolución en Tin Can, y la factibilidad de su integración con los distintos sistemas de e-learning pertenecientes a la Facultad de Informática de la UNLP.
- Implementar un Learning Record Store, LRS utilizando los estándares anteriormente mencionados.
- Implementar canales de comunicación entre el LRS y los LMS. Para el contexto de esta tesina, como caso de estudio se integrará la plataforma de gestión de cursos de cátedras de la Facultad de Informática (basada en Moodle).
- Desarrollar una aplicación para dispositivos móviles capaz de interactuar con el LRS.

1.2 Caso de estudio: Sistemas de aprendizaje de la Facultad de Informática

Se utilizará como caso de estudio los diferentes sistemas de aprendizajes que ofrece la Facultad de Informática de la UNLP. Se ha observado cómo los alumnos interactúan con los LMS que la facultad les provee, y se ha puesto especial atención en las fuentes externas de información a la cual los mismos acceden.

1.2.1 Caracterización de la problemática

Los alumnos de la Facultad de Informática no sólo adquieren conocimientos en las clases y en los sistemas de aprendizajes que la casa de estudios les provee, sino que a su vez se nutren de una variedad de fuentes de conocimientos externos al entorno de la facultad.

Esta situación se caracteriza por varias aristas: los alumnos tienen escasas posibilidades de hacer una devolución sobre los contenidos en los distintos sistemas de aprendizajes; los sistemas no comparten contenidos por falta de cooperación entre dichos sistemas de aprendizajes y finalmente la forma en que los alumnos acceden a la información. A continuación analizaremos cada una de estas aristas.

Posibilidades de hacer una devolución sobre los contenidos en los distintos sistemas de aprendizajes: los alumnos adquieren contenidos de los sistemas de aprendizaje y de una variada cantidad de sitios web, libros digitalizados y de sus propios apuntes. Es muy habitual para los alumnos consultar blogs, sitios con tutoriales, sitios con video tutoriales, y foros especializados. Toda esta información, que le ha sido útil, también puede serle de interés a otros alumnos. Los LMS tradicionales son unidireccionales en el sentido que los contenidos son proporcionados por el coordinador del curso y fluyen hacia los alumnos, nunca en forma inversa. Básicamente los LMS no permiten que los alumnos colaboren entre sí, ni con el profesor del curso.

Cooperación entre los sistemas de aprendizaje: los contenidos de una asignatura pueden ser complementados con el contenido de otra asignatura en el mismo LMS o bien puede ser enriquecido con contenidos que se encuentran en otro LMS, por ejemplo, el sistema de bibliotecas.

Actualmente los sistemas que posee la facultad no permiten:

- Extender contenidos que se encuentran en el mismo sistema de aprendizaje pero en otra asignatura.
- Complementar contenidos con contenidos pertenecientes a otro sistema de aprendizaje de la facultad.

Esto conlleva a que los contenidos se repliquen o que las referencias a otros LMS sean enlaces que por lo general no conducen a nada.

Formas de acceso a los sistemas de aprendizaje: los sistemas de aprendizajes son mayoritariamente accedidos por parte de los usuarios desde los navegadores web de las PC, esto porque a la hora de diseñar contenido y organizarlo se toman como referencias las resoluciones de una pantalla de PC y las capacidades que nos brinda un navegador web. Esto conlleva a que nos encontramos con que los sistemas tradicionales de aprendizajes no fueron ideados para ser accedidos desde dispositivos móviles. Acceder a los contenidos desde los navegadores web provistos por los teléfonos celulares resulta engorroso dado que la resolución del dispositivo no se adapta al sitio web. Aún cuando estos sistemas proveen un sitio web adaptado a móviles, la organización del mismo no suele estar orientada a los usuarios, teniendo éstos que hacer un gran número de clicks sobre las opciones para poder llegar al contenido deseado.

Capítulo 2

2.1 Tecnologías de la información aplicadas a la educación

Las instituciones educativas, a lo largo de su historia, han intentado ofrecerles a sus alumnos diversas formas de aprendizaje más allá del aula tradicional. Esta nueva forma de aprender que llamamos e-learning, es la unión de la ciencia de la educación, la cual aporta el contenido, y las ciencias informáticas, las cuales aportan el soporte tecnológico. Esta unión, que ha sido muy beneficiosa, ha ido evolucionando y refinando a lo largo del tiempo hasta lograr los estándares actuales.

2.1.1 Origen

Hoy en día, nos es impensado una clase en un aula sin un sistema de información que le de soporte, pero esto no siempre fue así. A lo largo de la historia, la ciencia de la educación se ha valido de los avances tecnológicos para poder llegar a los alumnos.

Los inicios de la educación a distancia se remontan al año 1728, donde el periódico la Gaceta de Boston, ofrecía a los interesados material de autoinstrucción, el cual era enviado a través del correo postal, con la posibilidad de contar con tutorías por este mismo medio.

La industrialización del siglo XIX y la demanda de mano de obra calificada después de la Segunda Guerra Mundial, favorecieron el surgimiento de ofertas formales de educación abierta, coincidiendo con la inclusión de la modalidad a distancia en universitarias. Esto da lugar a la proliferación de estudiar vía correspondencia como opción de aprendizaje.

Posteriormente con el auge de tecnologías vinculadas a la telefonía, la radio y la televisión le vuelven a dar un impulso a la educación a distancia, dando lugar a que los contenidos puedan ser difundidos en medios audiovisuales. Con el advenimiento de la computación, la computadora se convierte en un entorno multimedial interactivo, al sintetizar e integrar en sus programas texto, imágenes, sonidos y animación.

Luego con la llegada de Internet y el correo electrónico dan un nuevo significado y poder a la educación a distancia, generando la posibilidad de enseñar y aprender a través de la Red.

No obstante, hay que destacar que la utilización de las TIC para la educación y aprendizaje englobados bajo el término e-learning, no necesariamente implica una modalidad de formación a distancia. Su uso no sólo tiene como propósito superar la brecha

que implica la distancia entre alumnos y educadores. Es frecuente su empleo como apoyo a clases presenciales o incluso un modelo mixto de clases semi-presenciales lo que se ha denominado blended learning o b-learning.

2.1.2 Definición de e-learning

García-Peñalvo [15] define el concepto de e-learning de la siguiente manera: “se puede definir e-Learning como un proceso de enseñanza más aprendizaje, orientado a la adquisición de una serie de competencias y destrezas por parte del estudiante, caracterizado por el uso de las tecnologías basadas en web, la secuenciación de contenidos y actividades estructuradas según estrategias preestablecidas a la vez que flexibles, la interacción con la red de estudiantes y tutores y unos mecanismos adecuados de evaluación, tanto del aprendizaje resultante como de la intervención formativa en su conjunto, en un ambiente de trabajo colaborativo de presencia diferida en espacio y tiempo, y enriquecido por un conjunto de servicios de valor añadido que la tecnología puede aportar para lograr la máxima interacción, garantizando así la más alta calidad en el proceso”.

El e-learning consiste en la educación y capacitación a través de Internet. Este tipo de enseñanza online permite la interacción del usuario con el material de estudio mediante la utilización de diversas herramientas informáticas.

Este tipo de modalidad de aprendizaje ha transformado la educación, abriendo puertas al aprendizaje individual y organizacional. Es por ello que hoy en día está ocupando un lugar cada vez más destacado y reconocido dentro de las organizaciones empresariales y educativas.

El término "e-learning" es la simplificación de Electronic Learning. El mismo reúne a las diferentes tecnologías y a los aspectos pedagógicos de la enseñanza y el aprendizaje.

El e-learning comprende fundamentalmente los siguientes aspectos:

- El pedagógico, referido a la tecnología educativa como disciplina de las ciencias de la educación, vinculada a los medios tecnológicos, la psicología educativa y la didáctica.
- El tecnológico, referido a la TIC (Tecnología de la Información y la Comunicación), mediante la selección, diseño, personalización, implementación, alojamiento y mantenimiento de soluciones en donde se integran tecnologías propietarias y de código fuente abierto.

A primera vista, los componentes tecnológicos son los más tangibles y el ejemplo más significativo son las plataformas de e-learning o LMS que permiten la administración y

control de los aspectos administrativos de la capacitación, entre otras funciones. Los aspectos pedagógicos son el alma del e-learning y van a trabajar sobre los contenidos.

2.1.3 Ventajas y desventajas del e-Learning

En este apartado describiremos las ventajas y desventajas del e-learning, a las cuales suscribimos y que son enunciadas por Seoane Pardo y García Peñalvo :

Aspectos favorables

- Proporciona apoyo a las diferentes modalidades de formación: las plataformas de e-learning pueden utilizarse como soporte a cualquier modalidad de formación, por ejemplo como soporte a las clases presenciales.
- Ofrece una metodología centrada en el usuario: las plataformas de e-learning aportan al usuario autonomía para realizar su formación y determinar su ritmo de aprendizaje.
- Flexibilidad horaria: las plataformas de e-learning posibilitan la acción formativa en cualquier momento.
- Disponibilidad geográfica: las plataformas de e-learning permite acceder desde cualquier lugar con acceso a Internet.
- Reestructuración de la información: los contenidos en las plataformas de e-learning deben ser sintéticos, estar bien estructurados y centrarse en los parámetros principales, ofreciendo información adicional para profundizar, facilitando así el proceso de asimilación.
- Incremento de los índices de retención de contenidos: las plataformas de e-learning facilitan la retención de conceptos respecto a una clase presencial a través de la forma en que se estructura la información y el ritmo personal de aprendizaje.
- Contenidos más actualizados: las plataformas de e-learning facilitan la inclusión de formatos más visuales e interactivos para los contenidos, más adaptados didácticamente a las circunstancias. La actualización de los contenidos puede llegar instantáneamente a los destinatarios.
- Herramientas de interacción: las plataformas de e-learning ofrecen diversos medios de interacción entre los alumnos y con los tutores, entre ellos foros, chats y mensajería interna.
- Mayor razonamiento crítico y menor miedo al error: la falta de presencialidad, la posibilidad de usar medios asíncronos y una cierta sensación inicial de anonimato facilita la capacidad reflexiva y crítica en las plataformas de e-learning.
- Facilidad de inclusión de contenidos audiovisuales: las plataformas de e-learning, al estar soportadas por Internet, facilitan la inclusión de todo tipo de contenidos

audiovisuales, con la única restricción del ancho de banda y un balance entre calidad gráfica de los contenidos y rendimiento óptimo de la transmisión vía Web.

- Reducción de costos: permite reducir y hasta eliminar gastos de traslado, alojamiento, material didáctico, etc.
- Rapidez y agilidad: Las comunicaciones a través de sistemas en la red confiere rapidez y agilidad a las comunicaciones.
- Acceso *just-in-time*: los usuarios pueden acceder al contenido desde cualquier lugar, sólo necesitan una conexión a Internet.
- Flexibilidad de la agenda: no se requiere que un grupo de personas coincidan en tiempo y espacio.
- Reducción de costos: permite reducir y hasta eliminar gastos de traslado, alojamiento, material didáctico, etc.
- Rapidez y agilidad: Las comunicaciones a través de sistemas en la red confiere rapidez y agilidad a las comunicaciones.
- Acceso *just-in-time*: los usuarios pueden acceder al contenido desde cualquier lugar, sólo necesitan una conexión a Internet.
- Flexibilidad de la agenda: no se requiere que un grupo de personas coincidan en tiempo y espacio.

Aspectos negativos

- Pseudo e-learning: no debe usarse la plataforma de aprendizaje como simple “contenedor de materiales”. Es necesario explotar las potencialidades didácticas de las herramientas de interacción y disponer de una estrategia instructiva coherente y tutorizada.
- Falta de calidad de los contenidos: deben adaptarse al nuevo medio de difusión y ser coherentes con el diseño instruccional
- Desaprovechamiento de los recursos disponibles: debe combinarse el diseño de contenidos con las posibilidades de interacción de la plataforma de aprendizaje mediante un buen diseño instructivo para lograr buenos resultados de aprendizaje
- Resistencia al cambio: toda innovación produce cierta resistencia en los usuarios.
- Falta de estándares: desde el punto de vista tecnológico ya es posible crear contenidos estándar exportables a otros sistemas de formación. Sin embargo, “estandarizar” se entiende también como “crear un estilo propio” de la institución que acostumbre a los destinatarios a trabajar siempre con unos contenidos y procesos más o menos estándar

- Dependencia de la tecnología: el e-Learning genera dependencia tecnológica respecto a dispositivos y redes para acceder a la formación. Cualquier fallo o dificultad técnica, así como la falta de cualificación para el uso de las herramientas por parte de los usuarios, supone un obstáculo para el proceso

2.1.4 Uso de TIC en educación

Los principales participantes en el proceso de enseñanza mediante la web son: los profesores o tutores, los alumnos, los proveedores de contenido y los administradores.

Los profesores o tutores son los encargados de supervisar el proceso de enseñanza. Contrariamente a lo que muchas personas consideran, la enseñanza utilizando nuevas tecnologías no pretende sustituir a los profesores, aunque sí cambia su papel principal ya que pasa a ser más un dinamizador y un supervisor que un “transmisor” de conocimiento.

Los alumnos son los participantes centrales en el *e-learning*, ya que dependiendo de su rendimiento o satisfacción, se podrá evaluar el éxito de la enseñanza. Esta enseñanza necesita que el alumno tenga un papel activo y desarrolle mayores capacidades de autoaprendizaje, ya que la comunicación con el profesor y los compañeros es más limitada.

Los proveedores de contenidos educativos son responsables de la tarea de crear y diseñar el contenido, y de alguna manera del proceso de instrucción, de tal forma que se consigan los objetivos educativos pretendidos.

Los administradores del sistema se ocupan de gestionar los elementos de los catálogos de cursos, los horarios, los recursos, sesiones de aprendizaje, así como de los aspectos de seguridad y económicos.

2.2 Estándares

2.2.1 Definición de estándares orientados al e-learning

Por medio de la estandarización de los contenidos digitales buscamos que entre las plataformas tecnológicas se puedan establecer una serie de relaciones y competencias entre las mismas, las cuales permitan lograr interoperabilidad, reusabilidad, manejabilidad, flexibilidad, accesibilidad, durabilidad y escalabilidad de los contenidos digitales [16].

IEEE LOM [17][18] es un estándar versátil y flexible que permite la descripción de los recursos de aprendizaje a través de la especificación de sus metadatos. Si vamos un paso más allá, el sólo hecho de poder describir adecuadamente los recursos de aprendizaje resulta insuficiente. Para poder enseñar un concepto determinado se utilizan distintos recursos como pueden ser libros y actividades, también se necesita poder distribuirlos y

secuenciarlos de algún modo determinado para que el alumno siga un orden. Por ejemplo, no se puede comenzar con una actividad sin antes haber adquirido los conceptos teóricos que se presentaron en el libro; además un recurso puede ser útil para más de un propósito.

A diferencia con la educación presencial, en la que el alumno se apoya fuertemente en el docente, en la educación no tradicional las intervenciones del docente o tutor suelen ser escasas e incluso nulas en muchos casos. Los recursos educativos tienen un carácter educativo en virtud de su uso por alumnos en actividades educativas, y no por sus cualidades internas.

Se describirán cada uno de ellos, pero antes es necesario desarrollar un concepto que es fundamental para su correcta interpretación y entendimiento, que es el concepto de “metadato”.

2.2.2 Ventajas aportadas por los estándares en e-learning

Para este apartado nos hemos basado en lo aportado por Segura Escobar en su documento “Uso de estándares aplicados a las TIC en educación” [19], según el autor en *e-learning*, una de las principales funciones es facilitar el intercambio de los contenidos entre diversas plataformas y sistemas. Hay que evitar caer en el error de ver el estándar como un limitador de la iniciativa o creatividad personal.

Los contenidos estandarizados son aptos para su reutilización. Estos contenidos educativos reutilizables pueden ser de gran ayuda para simplificar el trabajo del docente, aunque lo utilicen en educación presencial o en *blended learning* o *b-learning*.

Dependiendo del rol del usuario, cada uno de ellos definirá desde su perspectiva lo que espera que cumpla el estándar. Estas necesidades de los usuarios respecto a los estándares se pueden agrupar en los siguientes grupos:

- **Interoperabilidad.** Que diferentes sistemas puedan relacionarse y cooperar entre sí de manera transparente, ya sea interactuando o compartiendo información.
- **Reusabilidad.** Que el contenido generado pueda ser utilizado fácilmente en un contexto distinto para el que fue creado.
- **Gestionabilidad.** Que el sistema pueda obtener y trazar la información adecuada sobre el usuario y el contenido.
- **Accesibilidad.** Que las diversas plataformas no impidan al usuario acceder al contenido correctamente y en el momento en que él lo requiera.
- **Durabilidad.** Que el contenido generado no quede restringido a una tecnología determinada y pueda ser reutilizado, sin que ello demande demasiada inversión, independientemente de la tecnología utilizada.

- **Escalabilidad.** Que el sistema pueda adaptarse y manejar su crecimiento sin que ello implique una pérdida de eficiencia o calidad.

2.2.3 Organismos participantes en la estandarización en e-learning

El concepto de e-learning y las tecnologías relacionadas a él son relativamente recientes, lo que provoca que el proceso de estandarización de las mismas sea una tarea compleja. Sin embargo, existen diversas iniciativas que están cooperando entre sí para acercarse cada vez más a una real estandarización de e-learning. A continuación se mencionan algunas de las más relevantes:

Aviation Industry CBT Committee (AICC): comité internacional relacionado al campo de la aviación especializado en el entrenamiento mediante el uso de sistemas informáticos [20]. La aviación es un campo apto para el desarrollo de simuladores y software de enseñanza. Uno de los aportes más relevantes realizado por AICC ha sido la recomendación para interoperabilidad o *Computer-Managed Instruction (CMI)*. Es una especificación sobre cómo crear contenido que se pueda comunicar con el mayor número de sistemas LMS.

Advanced Distributed Learning (ADL): iniciativa del Departamento de Defensa de EE.UU. y la oficina de Ciencia y Tecnología de la Casa Blanca, en colaboración con los departamentos de Educación y Trabajo. Su propósito es desarrollar herramientas para asegurar el acceso a materiales educativos de alta calidad que puedan ser adaptados a las necesidades individuales de cada usuario y que puedan distribuirse de forma sencilla [21]. ADL se ha centrado desde un principio en el aprendizaje sobre la Web y ha sido una de las organizaciones más activas respecto a la creación de estándares de tecnologías de aprendizaje. Como resultado de su trabajo se destaca un conjunto de especificaciones que, bajo el nombre de Shareable Content Object Reference Model (SCORM) (ADL SCORM, 2002, 2006), propone un modelo para la agregación, secuenciación y navegación de contenidos.

IMS Global Consortium: es una organización independiente [22], sin fines de lucro, que ha publicado una gran variedad de estándares de interoperabilidad relacionados con tecnologías de e-learning. Miembros de todas partes del mundo son encargados de liderar los grupos de desarrollo de estos estándares. IMS enfoca todos sus esfuerzos en promover y facilitar la adopción de tecnologías de aprendizaje innovadoras. Actualmente es uno de los

principales promotores y desarrolladores de especificaciones de código abierto orientadas al e-learning. La organización ha desarrollado especificaciones para cubrir diferentes necesidades del proceso de enseñanza, algunas referidas a los metadatos de los objetos educativos, al formato de empaquetamiento, distribución de cursos, entre otras.

Institute for Electrical and Electronic Engineers Learning Technology

Standards Committee (IEEE LTSC): es un comité de estandarización de las tecnologías aplicadas al aprendizaje, perteneciente al Institute of Electrical and Electronic Engineers (IEEE) [24][25], que se dedica a desarrollar estándares técnicos, prácticas recomendadas y guías para componentes software, herramientas, tecnologías y métodos de diseño que faciliten el desarrollo, implantación, mantenimiento e interoperabilidad de implementación de sistemas educativos. Sus mayores aportes están relacionados al estándar Learning Object Metadata (LOM), que es el estándar oficial que más se está utilizando actualmente en e-learning.

2.2.4 Estandarización: aspectos y procesos

Para que un estándar sea aceptado, su especificación debe estar basada en la recopilación de requisitos provenientes de múltiples fuentes. Es su nivel de aceptación lo que determina el éxito o no de un estándar. La obtención de un estándar formal se consigue como resultado de un proceso que puede dividirse en tres niveles de trabajo:

- *Nivel de especificación.* Primer paso del proceso. Se realizan recomendaciones basadas en el análisis de los requerimientos recopilados de los distintos participantes del proceso. Su fin es proponer la especificación a la comunidad e-learning para experimentar, corregir y actualizar en función de las nuevas necesidades detectadas.
- *Nivel de validación.* Se desarrollan nuevos productos incorporando la especificación elaboradas en el paso anterior y se realizan pruebas para valorar la efectividad y aplicabilidad de las mismas. Se analiza mediante modelos cómo las distintas especificaciones y estándares pueden ensamblarse para integrar un sistema e-learning completo.
- *Nivel de estandarización.* Una vez validadas las especificaciones, los organismos encargados de su estandarización, se encargan de refinarse y detallar qué requisitos satisfacen. Los productos que cumplen con la estandarización creada son acreditados. Notar la diferencia entre la especificación, que es un proceso de trabajo

en evolución, y el estándar acreditado, que es mucho más estable y menos propenso a cambios.

2.2.5 Objetos de aprendizaje

Los objetos de aprendizaje están íntimamente relacionados con los conceptos de accesibilidad y reusabilidad vistos anteriormente. Con el fin de lograr ambas es necesario que los contenidos estén identificados y sean fáciles de localizar.

Los objetos de aprendizaje identifican sus características a través de metadatos. Una de las normas más utilizadas para describir objetos de aprendizaje y sus propiedades es la norma Learning Object Metadata (LOM), la cual expresa los descriptores de los objetos utilizando lenguaje XML. Los atributos de los objetos de aprendizaje se identifican mediante etiquetas con el fin de que los sistemas pueden gestionarlos y procesarlos automáticamente, organizarlos, localizarlos, y reutilizarlos. Según LOM, se define un objeto de aprendizaje como “cualquier entidad, digital o no digital, que puede ser utilizada, para el aprendizaje, la educación o el entrenamiento”.

Esta es una definición genérica y ha hecho que se proporcionen otras definiciones más específicas como la de Wiley[26] que los define como “cualquier recurso educativo digital que pueda ser reutilizado como soporte para el aprendizaje”. Polsani[27] lo define como “una unidad didáctica de contenido, autocontenida e independiente, predispuesta para su reutilización en múltiples contextos instruccionales”.

2.2.6 Limitaciones de los estándares

Los estándares intentan cubrir la mayor cantidad de aspectos posibles de los sistemas de e-learning. Interoperabilidad, integración y reutilización son aspectos que se lograron cubrir de buena manera. En un escenario ideal, si hubiera estándares capaces de cubrir todos y cada uno de los aspectos de los procesos incluidos en los sistemas de e-learning, sería posible cambiar de LMS sin sufrir pérdida alguna de información.

Sin embargo, esta situación todavía no se ha alcanzado. Aún hay muchos aspectos que también son importantes en la enseñanza on-line, y que no están suficientemente resueltos. Por ejemplo, aportes realizados por los propios estudiantes en los foros de discusión, son difíciles de compartir entre distintos LMS o de migrar ante un eventual cambio de LMS.

2.2.7 Especificaciones y estándares con mayor aceptación

Como hemos descrito existen diversas iniciativas de estandarización, muchas de ellas trabajan en conjunto e incluso, en ocasiones, sus trabajos llegan a solaparse. Es difícil decir con exactitud cuáles están realizando un mejor trabajo. La única manera de poder medir el nivel de sus resultados es considerando cuán aceptados son los mismos por la comunidad e-learning.

Actualmente IMS Global Learning Consortium es el principal promotor y desarrollador de especificaciones abiertas que cubren más aspectos del e-learning. Asimismo, ADL y el desarrollo de su modelo de referencia SCORM y IEEE LTSC con su propuesta de metadatos para objetos de aprendizaje, son, junto a IMS, los que están teniendo una mayor repercusión en e-learning.

2.3 LMS: Learning management system

2.3.1 Definición

Un LMS o Sistema de Gestión de Aprendizaje [28] [29] es una aplicación instalada en un servidor, que administra, distribuye y controla las actividades de formación de una institución u organización.

Las principales funciones del sistema de gestión de aprendizaje son gestionar usuarios, recursos así como materiales y actividades de formación, administrar el acceso, controlar y seguir el proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros

2.3.2 Evolución

Inicialmente fueron diseñados para facilitar a los administradores y profesores la gestión y organización de cursos virtuales, relegando los aspectos pedagógicos a un segundo plano. Luego al combinarse estas herramientas con aspectos pedagógicos, fueron evolucionando a sistemas más amigables para todos los actores que participan en el proceso de aprendizaje, tomando como rol central el papel del alumno, así fue como los LMS pasaron a convertirse en prolongaciones o extensiones lógicas de los espacios de formación presenciales.

2.3.3 Propósito

Un LMS es una plataforma que ofrece y administra contenidos de instrucción. Su fin es la enseñanza y la formación para alcanzar objetivos tanto individuales como organizacionales. Permite realizar el seguimiento del progreso hacia el logro de esos objetivos y recoger y presentar datos para supervisar el proceso de aprendizaje de la organización en su conjunto. Un sistema de gestión de aprendizaje ofrece contenido, pero también se ocupa de la inscripción a cursos, administración de los cursos, seguimiento y presentación de informes, etc.

La mayoría de los LMS están basados en plataformas web para facilitar el acceso a los contenidos de aprendizaje y administración. Son utilizados por instituciones educativas para mejorar y apoyar cursos de enseñanza presenciales y ofrecer sus cursos a una población mayor de estudiantes. También pueden ser utilizados por las industrias y/o empresas privadas para capacitar al personal.

2.3.4 Ventajas

Entre las ventajas más importantes de los LMS que encontramos en los distintos entes educativos podemos destacar:

- Facilitan la presentación de información, la comunicación entre profesores y alumnos, la exposición de contenidos y materiales, el seguimiento del alumnado, etc.
- Permiten asignar roles entre los participantes a través del establecimiento de jerarquías.
- Favorecen la automatización de diversas estrategias de aprendizaje.
- Limitan el entorno donde el alumno desarrolla su proceso de aprendizaje.
- Presentan materiales que han sido adaptados, realizados o seleccionados por el docente en función de las necesidades y características que presupone en su alumnado.
- Generan feedback entre maestros y alumnos de manera casi inmediata.
- Permiten generar dentro de la propia plataforma un repositorio del proceso de aprendizaje.
- Controlan el proceso de aprendizaje de arriba hacia abajo (“top-down”).

2.3.5 Desventajas

A pesar de las las ventajas mencionadas, los LMS exhiben una serie de aspectos negativos respecto a los modelos de aprendizaje más recientes e innovadores; los

contenidos se encuentran encapsulados es decir, no aparecen en los motores de búsqueda y por consiguiente quedan relegados de la realidad; al mismo tiempo su accesibilidad es caduca, esto es, lo que se hace en el LMS queda en el sistema de gestión de aprendizaje hasta su cierre, sin posibilidad alguna de exportación.

Actualmente se cuestiona si los LMS constituyen un buen punto de partida en el momento de abordar y desarrollar el aprendizaje en las aulas virtuales debido a que su orientación eminentemente administrativa continúa vigente a día de hoy. Así lo corrobora el informe de la donde se señala que las funciones de los LMS se centran principalmente en aspectos administrativos, de gestión o burocráticos, observándose un limitado impacto de estos sobre los principios pedagógicos de la enseñanza.

2.3.5.1 Problema de trabajo colaborativo entre los alumnos

Las prácticas pedagógicas tradicionales, ubican al alumno como un mero consumidor de la información brindada por el profesor y el material que el o la institución provee al alumno para su formación. La comunicación se establece desde el profesor hacia los alumnos.

Las prácticas pedagógicas modernas ponen al educado en un rol central, sacándolo de la posición de un mero consumidor de la información y llevándolo a un área participativa dentro del aula. Este nuevo espacio que se le da al alumno, lo fomenta a colaborar tanto con sus profesores como con sus pares, fomentando así su interés con la asignatura y la responsabilidad con el curso.

Esta capacidad de colaborar entre pares, no es reflejada por los LMS actuales, los cuales están orientados los sistemas de enseñanza tradicionales e imposibilitan al estudiante a realizar sus aportes al curso al cual pertenece. Actualmente los LMS, solo cuentan con foros para hacer consultas, pero no permiten que los alumnos hagan sus aportes al curso.

2.3.5.2 Problema de los LMS para ser accedidos desde dispositivos móviles

Vivimos en un mundo hiperconectado, en el cual el modo en que accedemos a la red ha cambiado, ya no solo se accede a la misma mediante una computadora de escritorio desde una oficina u hogar, sino que podemos acceder desde nuestros teléfonos celulares y/o tablets desde cualquier lugar. Esto ha permitido que el acceso a la información se den en ámbitos donde antes era impensado, ya sea un una sala de espera de un consultorio, el buffet de la facultad, un micro, etc .

Los LMS son muy sencillos de usar desde una computadora de escritorio, pero no así desde un dispositivo móvil, esto provoca que los usuarios se choquen con un sistema que no está adaptado para ser accedido desde dispositivos móviles.

2.3.5.3 Dificultad de los LMS para interactuar con otros sistemas

Los LMS solo brindan al alumno los contenidos que el administrador del curso pone a su disposición, pero no le da la capacidad al administrador de asociar contenidos desde otros LMS. Por ejemplo, la facultad cuenta con un sistema de gestión de la biblioteca, y varios LMS los cuales tienen poca o nula interacción entre ellos.

Otra imposibilidad que tienen los profesores y administradores de los LMS es asociar información que se encuentra en sitios de Internet externos al LMS. Estos recursos que generalmente son videos-tutoriales, foros especializados o clases online de otras universidades; que tienen la capacidad de enriquecer un curso, no son contemplados por los LMS actuales.

Capítulo 3

En este capítulo se estudiarán los estándares de SCORM considerando sus pros y contras y cómo da lugar a su evolución hacia el estándar Tin Can. Haremos especial enfoque en Tin Can, por ser el mismo el que, según nosotros, abarca más ampliamente los nuevos requerimientos que este tipo de especificaciones deben tener.

3.1 SCORM: Sharable Content Object Reference Model

SCORM [30] es un conjunto de estándares y especificaciones que permite crear objetos pedagógicos estructurados. En esta sección estudiaremos su evolución y cuáles son las especificaciones que lo definen.

3.1.1 La especificación de SCORM

Los objetivos fundamentales identificados cuando la iniciativa ADL fue lanzada por el departamento de defensa de los Estados Unidos, fueron:

- Poder exportar cursos de un LMS a otro LMS.
- Reutilizar el contenido en distintos cursos.
- Secuenciar estos contenidos.
- Realizar búsquedas de contenido o repositorios a través de distintos LMS.

Muchas de las especificaciones creadas por otros organismos (entre ellos IMS, AICC, IEEE, etc) tenían la madurez suficiente pero no eran adoptados masivamente por los sistemas de e-learning. Estas especificaciones fueron agrupadas por ADL en el modelo SCORM con el objetivo de establecer un marco común para el aprendizaje asistido por computadora y basado en internet.

3.1.2 La organización de SCORM

La especificación de SCORM se centra en 3 aspectos, cada uno de éstos debidamente especificados en los siguientes documentos:

- Modelo de Agregación de Contenido (*Content Aggregation Model*, CAM): en este documento se describe los distintos tipos de objetos de contenido permitidos dentro de la especificación y detalla los mecanismos que se deben seguir para su empaquetamiento, descubrimiento en repositorios y su distribución e interoperabilidad entre distintos LMS

- Entorno de Tiempo de Ejecución (*Run-Time Environment*, RTE): este documento define el proceso de ejecución y los mecanismos de comunicación que tanto el LMS como el propio contenido deben emplear. La finalidad de este documento es garantizar que todos los LMS procesen los objetos educativos de manera similar y que el contenido sea intercambiable entre distintos LMS.
- Secuenciamiento y Navegación (*Sequencing and Navigation*, SN): este documento define los mecanismos para que los LMS puedan concatenar las actividades educativas de modo consistente. De tal manera que puedan secuenciarse de manera tal que se puedan navegar en un orden preestablecido para un curso.

3.1.3 Modelo de agregación de contenidos en SCORM

El modelo de agregación de contenido de SCORM representa el punto de partida a la hora de crear paquetes de contenido que pueden ser divididos en partes interoperables. Para ello se concibe un proceso en el cual se diseñan e implementan procesos educativos mediante la creación y agregación de recursos simples formando recursos educativos complejos que a su vez se organizan en una determinada secuencia. El modelo de agregación de contenido especifica los requisitos en los que se apoya este proceso definiendo:

- Un Modelo de Contenido: nomenclatura de los elementos que componen un proceso educativo.
- Un Modelo de Empaquetamiento: definición de cómo representar la estructura del contenido y cómo agregar distintos recursos educativos para su transporte entre distintos entornos
- Metadatos: un mecanismo para la descripción del contenido generado que permita realizar búsquedas y catalogaciones de este contenido
- Secuenciamiento: un modelo basado en reglas que describe el orden en que deben ejecutarse los distintos recursos educativos.

SCORM se compone de Assets, SCOs, Actividades, Organización de Contenido y la Agregación de Contenidos. Todos ellos se describen utilizando metadatos que permiten su búsqueda en repositorios de recursos y su potencial reutilización.

3.1.4 Entorno de Tiempo de Ejecución

El RTE es guía para lanzar contenidos y hacerle un seguimiento en un ambiente basado en Web. Para lograr esto el contenido educativo debe ser compartido a través de múltiples plataformas sin tener en cuenta las herramientas que se usen para crear o usar

los contenidos. Para que esto sea posible debe existir un método común para lanzar un contenido, un método común para que los contenidos se comuniquen con la plataformas y elementos de datos predefinidos que sean intercambiables entre la plataformas y el contenido durante su ejecución.

Los tres componentes del entorno de ejecución de SCORM son:

- Lanzador: es el mecanismo que define el método común para que las plataformas lancen un SCO basado en Web. Este mecanismo define los procedimientos y las responsabilidades para el establecimiento de la comunicación entre el contenido a mostrar y el LMS. El protocolo de comunicación está estandarizado a través del uso común de la API.
- API (Application Program Interface): proporciona un conjunto de funciones predefinidas para que la plataforma pueda comunicarse y controlar a los SCO que lanza. EL objeto queda enlazado a la plataforma cuando se lanza, enlace que se rompe cuando ya no se necesita el objeto. Las funciones también permiten que los objetos lean y escriban información en la plataforma y comprobar los errores que se produzcan durante el proceso.
- Modelo de Datos: está formado por una lista estandarizada de elementos (un vocabulario) que se emplean para intercambiar información.

3.1.5 Secuenciamiento y Navegación

Los recursos educativos deben presentarse al alumnado de manera tal que siga el orden establecido por los docentes, es por esto que es necesario secuenciar los recursos y agregar las reglas de navegación correspondientes.

- Secuenciación del contenido: Esta especificación describe cómo debe producirse la secuenciación de los contenidos almacenados en los SCO a través de una serie de eventos de navegación que pueden ser provocados por el estudiante o a iniciativa del propio sistema. Las posibles ramificaciones de los contenidos y los flujos que describen los posibles recorridos se establecen habitualmente durante el diseño. En esta especificación se define un método para representar el comportamiento de una actividad de aprendizaje, que queda recogido en el SCORM SN Model. La plataforma debe incluir las funciones necesarias para seguir la secuencia indicada en el SCO en tiempo de ejecución.
- Navegación del contenido: Las experiencias de aprendizaje se ofrecen a estudiante como una serie de actividades a desarrollar, identificadas por un secuenciador de contenidos. Los procesos de secuenciación que se implementa en la plataforma son componentes estáticos, que actúan como respuesta a un evento de navegación. La

navegación es el modo que tienen los usuarios finales y la plataforma de formación de identificar unas peticiones de navegación para realizar una experiencia de aprendizaje. Los dispositivos que se emplean para que el usuario introduzca las órdenes de navegación pueden pertenecer a la plataforma, a los objetos de contenido o a ambos.

3.1.6 Ventajas de utilizar SCORM

A lo largo de los capítulos se han estudiado el por qué la necesidad de un estándar de educación y el por qué de la creación de SCORM, es por esto ahora pasamos a enumerar las razones por las cuales dicho estándar es uno de los preponderantes en el mercado.

Entre las ventajas concretas podemos destacar [31]:

- **Accesibilidad:** dado que los contenidos se encuentran estandarizados, los mismos son distribuibles fácilmente desde y hacia los LMS .
- **Adaptabilidad:** algo posible con SCORM ya que podemos adaptar la formación en función de las necesidades de las personas y organizaciones. Pudiendo generar contenidos, actividades, evaluaciones, etc; para un sinfín de escenarios posibles .
- **Durabilidad:** los contenidos en formato SCORM, estarán disponible en todas aquellas plataforma que implementen es estandar.
- **Reusabilidad:** podemos utilizar diferentes componentes del curso para aplicar en diferentes contextos o situaciones.

3.1.7 Inconvenientes de utilizar SCORM

Más allá de las beneficios que nos aporta utilizar el estándar SCORM, el mismo presenta una serie de contras, algunas de las mismas producidas por los avances tecnológicos y otras por la propia evolución del aprendizaje. Estos son:

- El estándar SCORM fue pensado para navegadores web. Hoy en día contamos con gran cantidad de dispositivos móviles, para los cuales el estándar SCORM no fue pensado (por ser anterior a la proliferación de este tipo de dispositivos) y no se adapta a ellos correctamente.
- SCORM se acopla a un LMS, mediante un módulo específico para cada LMS. Este tipo de solución es ineficiente dado que se replican los módulos SCORM en distintos LMS, a su vez este tipo de implementación provoca un alto nivel de acoplamiento entre el LMS y el módulo SCORM.

- SCORM posee una definición de datos, mecanismos de comunicación y reglas del negocio muy compleja, lo que dificulta su implementación.
- El estándar SCORM es muy poco flexible a los nuevos tipos de fuentes de información externas al LMS's, tales como son las redes sociales, sitios web, blogs, etc.

3.2 Tin Can

Tin Can Api [32][33] es la evolución del estándar SCORM para hacer frente tanto a los nuevos cambios tecnológicos como así también a los cambios en la adquisición de conocimientos con base en e-learning.

Como hemos descrito en la motivación, es evidente que el modo en que las personas aprenden está cambiando, hoy los alumnos no sólo se nutren de contenidos mediante las clases en las aulas o los LMS utilizando su computadora, sino que consultan un gran número de fuentes de información desde distintos dispositivos electrónicos.

Es usual que los alumnos consulten a Wikipedia, blogs, foros, videos en Youtube, redes sociales e inclusive otros LMS en busca de los conocimientos deseados. No solo ha cambiado la forma en que las personas buscan información, sino también el modo de acceso a la misma. Hoy en día es común que se acceda a la información mediante dispositivos móviles, ya sean teléfonos celulares o tablets. Ni los LMS's ni el estándar SCORM dan soporte a esta nueva forma de aprendizaje.

3.2.1 Origen

Se lo llamó "Proyecto Tin Can" [34] porque estaba destinado a ser una conversación de dos vías, en alusión al sistema rudimentario que utilizaban los niños uniendo dos latas con un hilo, como muestra la Figura 1.



Figura 1: Sistema rudimentario de comunicación, que consta de dos latas unidas por un hilo.

Tin Can resuelve muchos problemas que las especificaciones anteriores sufrieron, para esto añade nuevas capacidades, nuevos casos de negocios, y nuevas formas de gestión de contenidos.

3.2.2 Característica que agrega a SCORM

Al igual que SCORM, Tin Can API registra actividades del usuario. La diferencia es que registra un abanico más amplio de actividades y esto nos podría aportar más información sobre un proceso de aprendizaje [34][35]. SCORM registra si el usuario ha iniciado o no un material dentro de un curso o cuál es el porcentaje de avance dentro de un periodo de tiempo, es decir interacciones que se producen dentro de un LMS; Tin Can API, en cambio, puede registrar interacciones que se producen tanto dentro como fuera de la plataforma de aprendizaje. De hecho, entre los ejemplos de aplicación hay algunos juegos.

Entre los datos alternativos que puede capturar se destacan:

- Las interacciones que se realizan en dispositivos móviles.
- Las actividades en simulaciones, mundos virtuales, juegos.
- Los libros leídos por una persona.
- Las actividades informales que se realizan en redes sociales.

La Figura 2 muestra como un LRS puede interactuar con distintos sistemas informáticos.

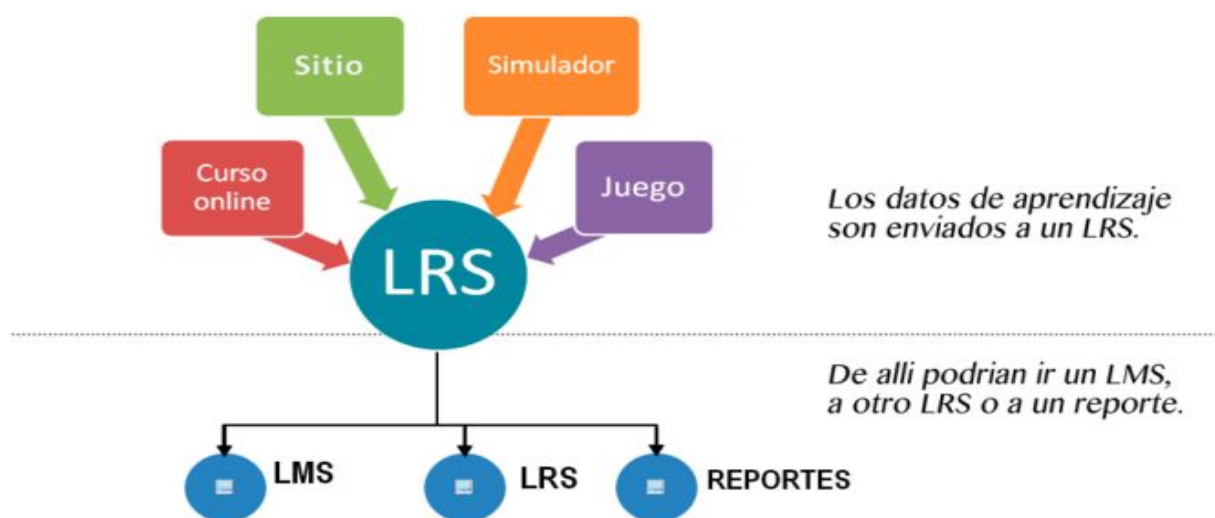


Figura 2: Interacción entre LRS y diversos sistemas. Extraído de Tin Can Api[35]

3.2.3 ¿Cómo funciona Tin Can API?

Todos podemos aprender en cualquier momento, medio o lugar, y Tin Can API nos permite crear una actividad, grabarla y registrar cuándo y dónde se produjo el aprendizaje. Mediante la utilización de los Learning Record Store (LRS) es que Tin Can puede lograr dicho registro.

Un LRS nos permite identificar quién, cómo y qué actividad se ha realizado, especificando un “Sujeto”, un “Verbo o acción” y un “Objeto de aprendizaje”. Ejemplo: Yo envíe un e-mail.

Los LRS son independientes de los LMS, ya que las actividades pueden producirse dentro o fuera de un plataforma de formación. En la figura 3, vemos cómo se relacionan el alumno con las experiencias educativas.



Figura 3: Capas de Tin Can. Extraído de la documentación de Tin Can[33]

3.2.4 ¿Cómo se estructura Tin Can?

El standard Tin Can API se estructura en 4 capas:

Capa nivel 1: en esta capa de Tin Can API se ha desarrollado una versión nueva y optimizada del estándar SCORM. Permitiendo:

- Realizar el seguimiento en entornos desconectados o semi conectados, es decir, desde cualquier dispositivo (smartphones, tablets, gps, etc..) se puede generar contenido para ser subido cuando se restablece la conexión a internet.
- Lanzar contenidos fuera de los LMS.
- Mantener un control total sobre la distribución de contenidos y seguimiento del usuario.
- Guardar múltiples resultados para un curso (test inicial, test final, etc)
- Permitir múltiples intentos.
- Resultados detallados de las pruebas.

Capa nivel 2: en esta capa se ha desarrollado la tecnología necesaria para poder grabar cualquier actividad de aprendizaje informal. Es decir, cualquier actividad relacionada con las

redes sociales (Facebook, Twitter, LinkedIn, Youtube, etc), uso de recursos Web 2.0 (Blog, Wiki, etc), actividades con dispositivos móviles (Smartphones, tablets, etc) y actividades fuera de la red.

Capa nivel 3: en esta capa se ha desarrollado el acceso de lectura y escritura de los datos posibilitando que el LRS se comunique con otras herramientas de análisis, de generación de informes e incluso con otros LRS, de forma que la exportación de datos se realice de forma sencilla y permita el análisis de contenidos y de resultados. Tin Can exige que los LRS contengan datos accesibles, permitiendo tanto la grabación de datos, como su lectura.

Capa nivel 4: permite analizar la efectividad de la formación realizada con relación al desempeño de la persona en el puesto de trabajo, mediante el seguimiento de la actividad que realiza la persona tanto en redes sociales como en la propia empresa, analizando después la convergencia entre ambas actividades, de forma que en un futuro se pretende poder discernir qué itinerario de aprendizaje es el más adecuado para cada persona.

3.2.5 Núcleo de Tin Can

El núcleo de Tin Can API es un sentencia con una estructura bastante simple:

Actor, Verbo, Acción

Los siguientes son ejemplos de sentencias Tin Can:

| Sujeto | Verbo | Objeto de aprendizaje |
|----------|---------------|---|
| Pedro | Aprobó un 95% | Actividad redacción orientada a recursos transmedia |
| Juan | Ingresó | Curso de Competencias comunicativas |
| Mauricio | Reprobó | Wiki sobre la comunicación en entornos de aprendizaje |
| Javier | Leyó | Valoración de conceptos y experiencias e-Learning |
| Pablo | Participó | Foro de inducción al curso de comunicación y experiencias |

Estas declaraciones pueden ser simples o complejas. Los actores, verbos y objetos pueden variar ampliamente, y pueden ser descritos con diferentes niveles de detalle. Los actores también pueden describirse de varias maneras diferentes. Un actor no tiene que ser un estudiante, puede ser un instructor que está afirmando una declaración, o incluso un agente de software. Es responsabilidad del usuario final decidir el nivel de complejidad que se necesita.

Estas declaraciones serán los resultados de experimentar parte del contenido de e-learning; tomar una prueba, completar una simulación, o incluso un ensayo calificado por un instructor. El contenido ya no sólo se limita a SCO, y en este sentido el contenido se

convierte en parte de algo más grande que llamamos "actividades". De esta manera, podemos pensar a los creadores de contenido como "proveedores de actividades."

3.2.6 Capacidades de Tin Can

Tin Can presenta una serie de características [36] que hacen de él, un estándar de e-learning moderno y completo, a continuación pasamos a detallarlas.

3.2.6.1 Simplicidad

Los estándares de e-learning tradicionales son complejos y su implementación es costosa y lleva un tiempo significativo. Con la intención de simplificar la implementación Tin Can mejoró varios aspectos en comparación con su antecesor SCORM:

- Ya no se tiene que lidiar con Javascript.
- La complejidad de las actividades y la comunicación con el LRS fue reducida (menos llamadas a la API.)
- Ya no hay archivos de manifiesto ni estructuras de archivos tan complejas.
- No existen requisitos complejos para las actividades. Sólo tiene que ser capaz de enviar la declaración "Hice esto."
- Dado que la actividad se ejecuta en su propio entorno (fuera de los LRS) no hay requisitos de interfaz de usuario que deba cumplir la actividad o el LRS.
- Las declaraciones de Statements se han vuelto muy sencillas, de la forma Actor + Verbo + Actividad.

3.2.6.2 Independiente del navegador

La API de Tin Can elimina la necesidad de un navegador de Internet. Esto posibilita que el contenido sea mucho más diverso y que los usuarios tengan múltiples formas de acceso. La creación de aplicaciones móviles, simuladores y juegos que se ajusten a la API es más sencilla. Solo tienen que enviar statements al LRS.

3.2.6.3 Distribución de Contenido

Un inconveniente de las especificaciones tradicionales de e-learning ha sido siempre la capacidad de alojar contenido fuera de los LMS. Hay algunas soluciones, pero ninguna de ellas resuelve el problema correctamente como lo hace la API Tin Can.

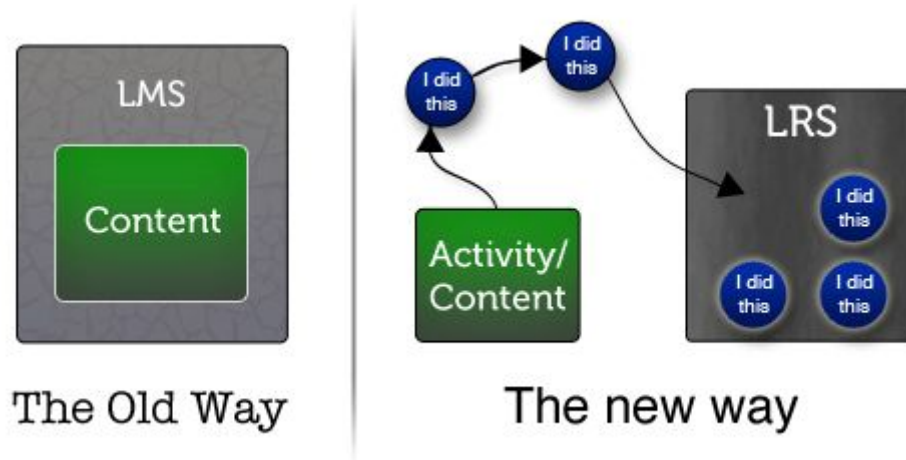


Figura 4: Comparación entre LMs y LRS. Extraído de la documentación de Tin Can [33]

Como vemos en la Figura 4, para Tin Can la actividad vive fuera de los LRS lo cual da lugar a los siguientes escenarios:

- No hay necesidad de importar cursos del LMS.
- Los creadores de contenido, proveedores de actividad, tienen un control completo sobre su contenido.
- Cuando una actividad se actualiza, es decir, corrigió un error tipográfico, reformulado una pregunta, no hay necesidad de enviar los cambios a cada LMS.
- Las actividades ya no tienen que estar almacenadas en el mismo repositorio. Un proveedor actividades puede componer contenido desde diferentes ubicaciones.
- Una actividad no solo puede ser "contenido", también puede ser cualquier evento de aprendizaje.

3.2.6.4 Acceso a la información

La API de Tin Can permite que todo se registre en el LRS y que el acceso a la información sea sencillo. Esto abre un nuevo nivel de detalle en los reportes que pueden realizarle a partir de la información registrada.

Los LRS pueden consultar por el actor, verbo u objeto que quiera o cualquier combinación de ellos. La cantidad de detalles que se puede consultar sólo está limitado por la complejidad de las declaraciones que el generador de actividad opta por generar.

Hay oportunidad de almacenar una cantidad significativamente mayor de datos que antes. Dependerá del LRS determinar diferentes niveles de usuario y a qué datos accede cada usuario. Hay un montón de cosas que se pueden hacer con estos datos que antes no

se podía. Un ejemplo sería la de obtener informes sobre cuestiones o sectores específicos, en lugar de sólo un curso entero.

3.2.6.5 Variables definidas por el usuario

Una variable definida por el usuario es información dinámica que vive dentro del contenido. Hay dos maneras de incorporar variables definidas por el usuario. La primera es donde la información externa se introduce en el LRS para ser utilizada por una actividad. Algunos ejemplos de cómo esto puede ser usado:

- Un curso sobre el mercado de valores que arroja información de las acciones en tiempo real.
- Un simulador de vuelo que utilice velocidades de viento del mundo real.
- Un curso "on-boarding" para una empresa que aporta datos variables como nombres de personal, programación de vacaciones, etc.

La otra manera de utilizar variables definidas por el usuario es mediante el almacenamiento de información sobre un alumno o una actividad como una variable que se puede utilizar más adelante en la presentación de informes. Estas variables pueden ser cualquier dato arbitrario como los números de mercado de valores o la velocidad del viento a partir de los ejemplos anteriores.

3.2.6.6 Aprendizaje no inicializado en el LMS

La naturaleza de la API Tin Can implica que el aprendizaje no tiene por qué empezar en un LMS. De hecho, el LRS ni siquiera necesita tener conocimiento previo de qué actividad o actor está involucrado. La actividad se limita a enviar un statement al LRS una vez que se ha completado una actividad. Los LRS entonces registra el statement y sus datos de acuerdo con lo que el creador de la actividad ha decidido enviar. Esto nos permite estudiar casos de usos nuevos:

- Los usuarios pueden iniciar su propio aprendizaje y reportarlo al LRS.
- Cualquier cosa en Internet puede ser reportado al LRS como experiencia de aprendizaje, no sólo cosas asignados a un usuario o grupo.
- Los instructores pueden enviar las actividades directamente al usuario sin pasar por el LMS.

3.2.6.7 Juegos y Simulaciones

Simulaciones y juegos se han utilizado en el e-learning desde hace tiempo, y el seguimiento y la información en gran parte se ha limitado a los sistemas propietarios en las

que viven. La API Tin Can permite que las simulaciones y juegos puedan ser rastreados en cualquier LRS. Esto amplía la competencia y abre más posibilidades a los creadores de contenido, las instituciones de enseñanza, y los alumnos. La API Tin Can permite:

- Contenido distribuido: simulaciones y juegos pueden vivir localmente o en sus propios servidores y enviar declaraciones a un LRS externo.
- Múltiples alumnos y aprendizaje en equipo: útil para juegos multijugador y simulaciones. Los datos pueden restringirse a un grupo y mostrar información importante para el grupo.
- Las actividades pueden vivir fuera del navegador, un factor decisivo para la mayoría de las simulaciones y juegos. Además, dado que las actividades no se limitan a una sesión del navegador muchos más datos pueden enviarse al LRS.
- Interacción con el instructor: útil para cuando un instructor quiere intervenir durante una simulación, o si un instructor quiere cambiar el escenario en una simulación en marcha. Si bien no hemos definido directamente una manera para que esto suceda, se puede hacer mediante el uso de variables definidas por el usuario.
- Variables definidas por el usuario: es donde realmente se puede ser creativo. Utilizar la velocidad actual del viento en tiempo real en un simulador de vuelo, utilizar datos del mundo real desde casi cualquier fuente para crear simulaciones, juegos, etc..
- Especificaciones de e-learning antiguas tienen problemas que son inherentes a la forma en que los juegos y las simulaciones son registrados. Juegos y simulaciones a menudo son pensados para ser ejecutados y registrados por largos períodos de tiempo y con muchos intentos de la misma actividad. Tin Can avanza sobre este área donde las otras especificaciones de e-learning suelen fallar.

3.2.6.8 Seguimiento de actividades en el mundo real

Los sistemas de e-learning tradicionales registran acciones digitales como cuando un curso se ha completado. ¿Qué pasa si se quiere registrar actividades del mundo real, no necesariamente de e-learning, como la participación en clase? La API Tin Can ofrece métodos para integrar las actividades del mundo real con el aprendizaje digital.

Ejemplos de actividades del mundo real que pueden ser registradas:

- Cualquier información que pueda ser introducida manualmente en una grabadora.
- Que se asistió a una conferencia.
- La asistencia a clases.
- La participación en clase.

La implementación más simple de esto es que un instructor notifica al LRS que el usuario ha completado una cierta actividad.

3.2.6.9 Seguimiento Desconectado

La mayoría de las anteriores especificaciones de e-learning requieren una conexión constante para seguir el contenido. El contenido puede ser suspendido y reanudado luego, pero no puede ser experimentado sin conexión. El LMS puede proporcionar una versión offline de su API para que el usuario pueda experimentar el contenido sin conexión, pero requiere trabajo adicional para el LMS y no todos los LMS ofrecen esta característica.

Una parte del contenido o actividad que utiliza la API Tin Can puede ser lo suficientemente inteligente como para detectar si tiene o no tiene una conexión de red. Si no hay conexión y el contenido o actividad tiene su propio almacenamiento, puede almacenar datos de forma local y transmitir de nuevo al LRS cuando haya conexión.

Tin Can es ideal para el seguimiento de contenidos de larga duración, donde es poco práctico mantener una sesión SCORM abierta durante días o incluso semanas.

3.2.6.10 Cambios de plataforma

La API Tin Can permite que un alumno pueda iniciar una actividad en una plataforma, como una computadora en la casa, y continuar esa actividad en un dispositivo diferente, a través de una aplicación en un teléfono móvil por ejemplo. Lo que hace posible esta transición entre plataformas es que el contenido no se almacena en un LMS. El desarrollador de la actividad puede crear diferentes versiones de la misma para cada plataforma en la que que desee utilizarla. Tin Can permite a un LRS ver las actividades de aprendizaje en diferentes dispositivos como una sola actividad. Esto permite que un alumno transite de un dispositivo a otro con la misma actividad y que se reporte para trabajar en toda la actividad o que se desglose para ver dónde y cómo experimentó la actividad. Para lograr esto la API permite guardar el estado de la actividad y compartir este estado a través de las diferentes plataformas.

3.2.6.11 Interoperabilidad

Tin Can hace hincapié en la capacidad de interoperabilidad, solucionando muchos problemas de interoperabilidad de las especificaciones actuales de e-learning que pasamos a describir:

- Problemas de interfaz de usuario: los LMS tienen sus propias interfaces de usuario. Al crear contenido que sea compatible con el mayor número de LMS como sea

posible, no siempre se pueden tener en cuenta todas las variantes de interfaces de usuario que los LMS presentan. Tin Can delega la interfaz de usuario a los contenidos / actividades. De esta manera, los creadores de contenidos / actividades pueden estar seguros de que su contenido será experimentado como se pretende.

- Submit del contenido: en Tin Can, que un alumno realice el “envío” o “presente” una actividad significa que ha terminado, o está listo para entregar algo, o está listo para ser evaluado. SCORM no ofrece un concepto de "presentado" o "no presentado". En SCORM, el contenido se tiene que suspender el mismo para evitar la pérdida de datos, y cuando el contenido ha sido completado y suspendido, el LMS asume que está listo para ser evaluado. Pero ¿qué pasa si un alumno quiere volver y revisar sus respuestas sin que el LMS asuma que está listo para ser evaluado? Tin Can elimina todos los supuestos que un LMS puede hacer en la zona de finalización.

Este problema se resuelve mediante la incorporación de la noción de "presentado" o "no presentado". Esto permite a los LMS saber si un alumno ha completado el contenido y está listo para enviarlo, o si han completado el contenido y quieren volver y revisarlo antes de presentarlo para su evaluación.

3.2.6.12 Secuenciación

Tin Can elimina totalmente los problemas de secuenciación del LMS que caracteriza a SCORM. No hay más problemas de interoperabilidad con los LMS ya que éstos no se encargan de la secuenciación.

3.2.6.13 Interacción entre alumnos e instructores durante el entrenamiento

El e-learning ha sido una experiencia muy solitaria. Uno toma un curso y luego es evaluado. Tin Can abre nuevas posibilidades y permite que los desarrolladores de actividades creen nuevas formas de interacción entre los alumnos y los instructores.

¿Cuáles son los casos de uso?

- Un profesor puede evaluar un ensayo y dejar comentarios y / o sugerencias durante la evaluación.
- Un instructor puede observar una simulación y ajustar el escenario sobre la marcha.
- Los comportamientos de los alumnos pueden ser observados más de cerca, lo que permite la optimización de contenido.
- Los instructores pueden responder a las inquietudes de los alumnos en tiempo real sin tener que dar una calificación.

- Los profesores pueden dejar comentarios durante los cursos para aclarar confusiones que los alumnos puedan tener.
- Chat entre los instructores y los alumnos durante las actividades.

Cabe señalar que las actividades / contenidos necesitan proporcionar las capacidades anteriores para poder ser registrados en el LRS.

3.2.6.14 Aprendizaje colaborativo y aprendizaje basado en el equipo

Aprendizaje colaborativo en Tin Can significa que pueden haber varios estudiantes que experimentan la misma actividad. Ellos no tienen que estar físicamente juntos ni deben experimentar la actividad al mismo tiempo aunque Tin Can lo permita. Por medio de Tin Can los alumnos pueden:

- Hacer preguntas a otros alumnos.
- Dejar comentarios para otros alumnos y / o el creador de la actividad.
- Opinar sobre respuestas de otros alumnos.
- Ser evaluados en relación a su contribución al grupo.

Todas estas interacciones pueden ser registradas como declaraciones enviadas a un LRS. Este tipo de aprendizaje colaborativo era simplemente imposible con anteriores especificaciones de e-learning.

El aprendizaje basado en equipo permite el seguimiento de los resultados de las experiencias que se tienen no solo por una persona sino por más de una.

Declaraciones como "El equipo 3 de Laboratorio de Software ha completado el trabajo práctico 1" es un ejemplo válido. Esto es útil cuando es importante hacer un seguimiento e informar sobre el rendimiento del equipo, así como las personas que lo conforman.

3.2.6.15 Etiquetar contenido con capacidades

Actualmente, Tin Can permite:

- Consultar y filtrar información basándose en etiquetas. No está definido actualmente en la API de Tin Can, pero se encuentra en proceso de implementación.
- Consultar y filtrar información basándose en etiquetas favorece la organización. En el pasado no había una manera estandarizada de organizar el contenido basado en diferentes habilidades.
- Etiquetar el contenido / actividades con habilidades permite a los administradores de LRS y estudiantes encontrar y gestionar el contenido que necesitan fácilmente. Esto

es particularmente útil para organizaciones grandes o con una gran cantidad de contenido.

3.2.7 Debilidades de Tin Can

La naturaleza de añadir nuevas funcionalidades a algo tan trascendental como las especificaciones de e-learning implica inherentemente compensaciones de diseño. Enfocarse en un área a menudo implica necesariamente renunciar a otras áreas.

3.2.7.1 Acuerdo de definición de actividad

¿Cómo nos aseguramos que una actividad no es referenciada dos veces de dos maneras diferentes?, ¿Cómo nos aseguramos que dos actividades diferentes no sean referenciadas de la misma manera? Estas son preguntas que tienen que plantearse cuando hablamos de actividades como parte de un statement.

Idealmente, las organizaciones tomarían el control sobre la nomenclatura de las actividades. Los nombres de dominio pueden ser utilizados para indicar que un statement se originó a partir de una fuente, y los statements podrían tener que ser más complejos con el fin de asegurarse de que son lo más preciso posibles.

URLs y URIs se pueden utilizar para especificar quien realmente posee un recurso y de dónde viene, pero quien controla el dominio utilizado en una URL tendría que asegurarse de hacerse cargo de no tener nombres duplicados para distintos recursos.

3.2.7.2 Variación de verbos

¿Cómo establecemos el estándar para los verbos que se utilizan en los statements?, ¿Una prueba se “completa” o “termina”? ¿Acaso Juan “leyó” el capítulo 1 o “estudio” el capítulo 1? En el contexto del e-learning, “completado” y “terminado” son prácticamente la misma cosa, pero, a efectos de reportes, son completamente diferentes. Tin Can es capaz de agrupar ciertos verbos que significan lo mismo y asignar diferentes resultados lógicos para ellos.

3.2.7.3 Complejidad de informes

La complejidad de los informes está ligado al problema de la variación de los verbos. Mientras los verbos permanezcan estandarizados la complejidad de los informes será un problema menor. Si “experimentar” y “leer” significan lo mismo para un LRS, entonces cualquier persona que realice un informe para ver qué alumnos han “leído” el capítulo 1

verán tanto a los alumnos que han “leído” el capítulo 1 como a los que han “experimentado” el capítulo 1.

Si un alumno quiere saber qué nota obtuvo en una prueba, eso es todo lo que se debe informar, a pesar de que hay muchos más statements detrás de esa nota, por ejemplo "el alumno completó la prueba" y "el instructor calificó la prueba". No es necesario informar los statements adicionales, pero podrían ser útiles si el instructor quisiera una lista de los alumnos que él ha evaluado.

3.3 LRS: Learning record store

Un LRS [37] es una implementación concreta del estándar Tin Can, el cual es un repositorio de registros de aprendizaje. En él se almacenan las actividades realizadas por los usuarios, en este caso por los alumnos de la Facultad de Informática de la UNLP, y su progreso en dichas actividades. Todas las actividades generadas por Tin Can son enviadas a un LRS. Los registros de aprendizaje almacenados en un LRS, pueden ser accedidos por un LMS, por herramientas generadoras de reportes, otro LRS y aplicaciones móviles.



Figura 5: Consumidores/Productores de información de un LRS

Tin Can permite a las actividades de aprendizaje generar sentencias o registros de aprendizaje de la forma “Yo hice esto” o “Actor Verbo Objeto”. Estas sentencias son enviadas al LRS (figura 5). La principal función del LRS es almacenar y recuperar la información de las sentencias generadas por Tin Can.

Con los LMS tradicionales la información generada por los usuarios quedaba atada

al LMS y a la organización que administraba el LMS. Con la introducción de los LRSs, compartir información de aprendizaje es ahora posible y dicha información es capaz de acompañar al usuario responsable de generarla a donde quiera que él vaya.

3.3.1 Funcionamiento del LRS

Analizaremos como es el funcionamiento del LRS, para lograr ésto hemos hecho hincapié en las especificaciones técnicas del LRS [38][39][40] .

3.3.1.1 Registrando experiencias educativas

El flujo de trabajo para registrar una actividad educativa es el siguiente: un alumno tiene una experiencia de aprendizaje, la misma puede ser en un curso online, en el trabajo, en un juego, etc., dicha experiencia es lanzada al LRS por medio de alguna aplicación que interactúe con nuestro LRS y éste registra la experiencia de aprendizaje con todo su contexto.

Dicha experiencia de aprendizaje ahora queda disponible para el alumno creador de la experiencia y para los actores/grupos autorizados, como lo ilustra la figura 6.

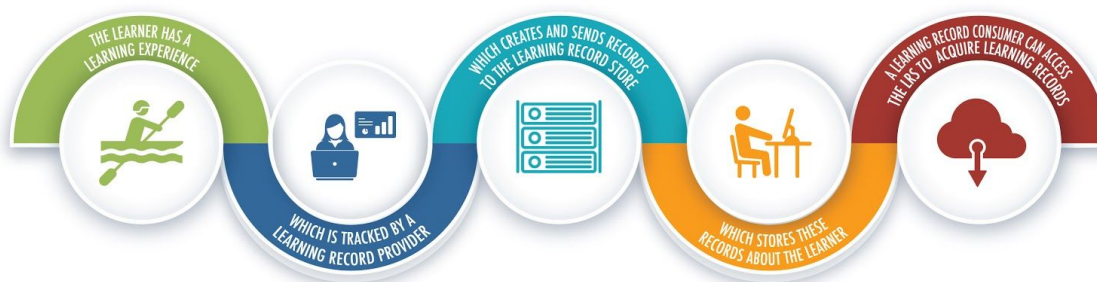


Figura 6: Registro de experiencias educativas. Extraído de la documentación de Tin Can[39]

3.3.1.2 Actividades en el LRS

Comprender cómo se define y describe una Actividad, unívocamente identificada por su IRI, es un concepto clave de xAPI. La Figura 7 muestra este proceso. Una Actividad, como parte de un Statement, posee metadatos, los cuales pueden ser definidos dentro del mismo Statement en la definición de la Actividad. El id de cada actividad es un Internationalized Resource Identifiers (IRI) el cual también puede contener metadatos. El Proveedor de Metadatos (Metadata Provider) es el encargado de resolver correctamente los IRI, tanto sus metadatos como el IRI en sí.

xAPI Statement

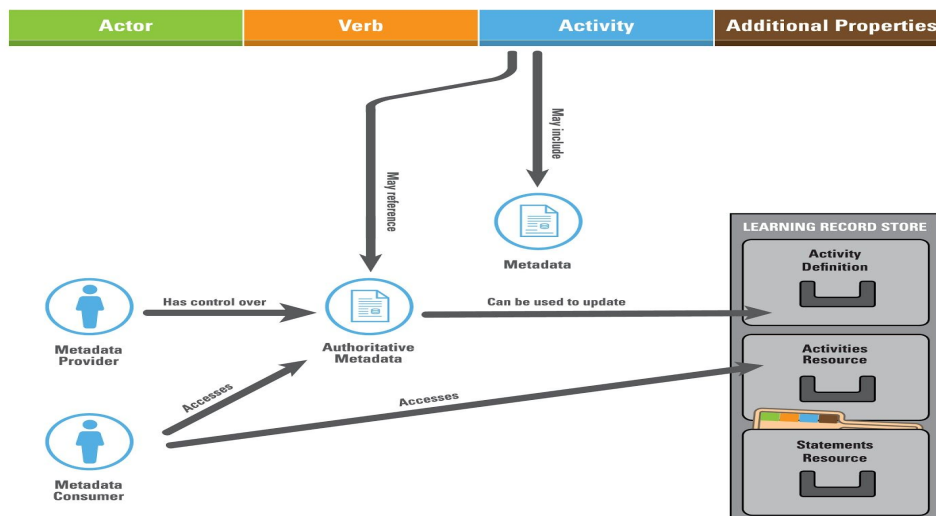


Figura 7: Recupero de experiencias educativas. Extraído de la documentación de Tin Can[39]

3.3.1.3 Acceso de usuarios

Tin Can nos permite el acceso selectivo a datos personales. Esto se hace a través de la gestión de lo que se llaman agentes. En Tin Can cada agente está representada por el actor en el statement, es el "yo" en "yo hice esto". Cada Agente o Grupo en xAPI corresponde a una persona. Un alumno que envía Registros de Aprendizaje a un LRS podría tener varios agentes asociados a él.

Un alumno accede a múltiples servicios. Algunos de estos servicios se utilizan en el trabajo, otros en el hogar. Algunos se usan para propósitos sociales y otros se usan para propósitos educativos o profesionales. Por lo tanto, hay múltiples personas funcionando colectivamente dentro de estos servicios. Cada uno de estos servicios envía datos al LRS. Posteriormente, hay declaraciones de tres personajes diferentes del mismo alumno.

El LRS puede agregar toda la información de cada uno de los personajes en un objeto "Persona". Este objeto se puede recuperar desde el LRS a través del recurso Agentes. Cómo el LRS sabe que estos múltiples personas pertenecen a una sola persona está fuera del alcance de esta especificación y hay varios enfoques diferentes que un LRS puede tomar. También es posible que algunos LRS no tengan ningún mecanismo para asociar personas.

Lo anteriormente descrito puede observarse en la figura 8

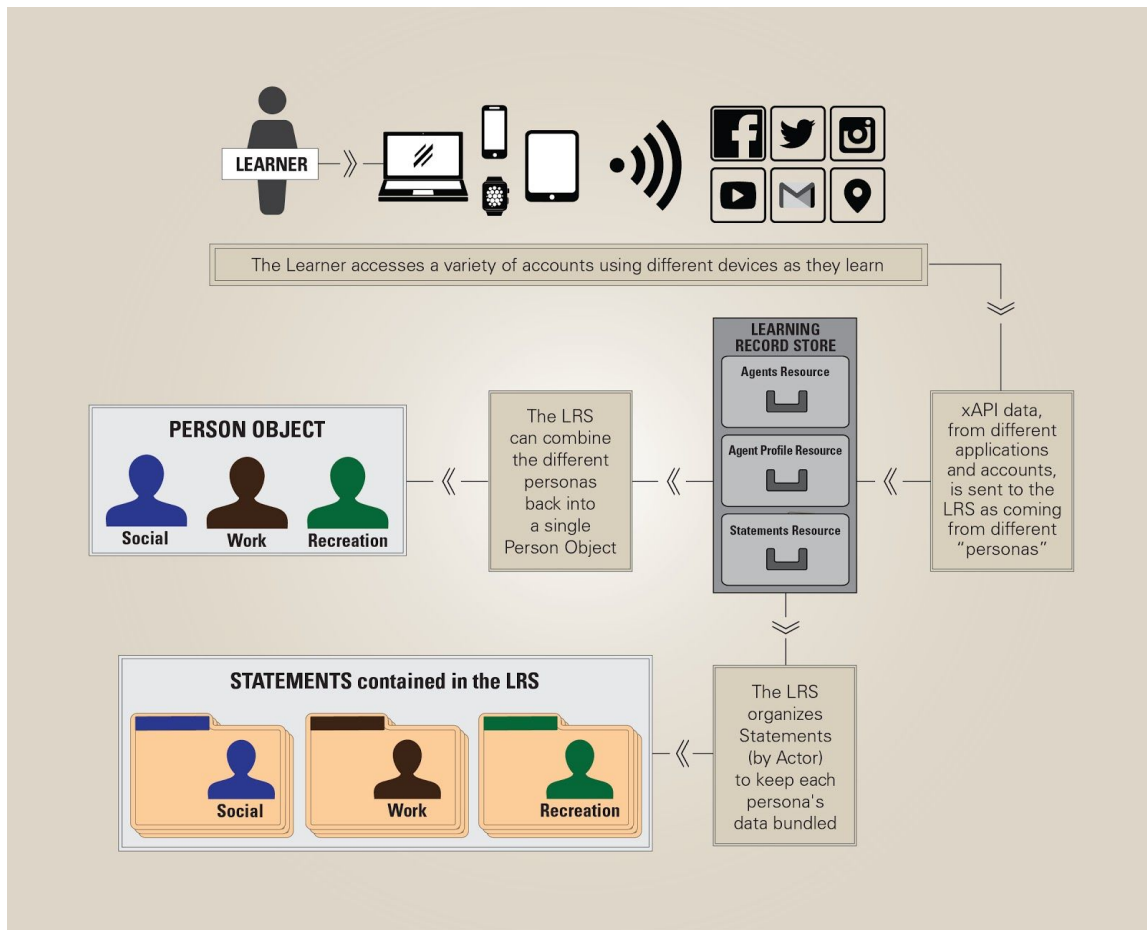


Figura 8: Manejo de grupos en el LRS. Extraído de la documentación de Tin Can[39].

3.3.2 APIs especificadas por el estándar Tin Can

El estándar Tin Can es una colección de cuatro APIs RESTFul [41]. La *Statement API* está centrada en la generación de sentencias y su transferencia entre LRS y otras tres, denominadas *Documents APIs* (State, Activity y Agent API), no trabajan con sentencias, están dedicadas al almacenamiento de información más compleja (documentos Word, videos, imágenes, etc.). Para construir un LRS debemos implementar las cuatro APIs [42].

- **Statement API:** mediante la Statement API se realiza el intercambio de Statements desde y hacia el LRS. Provee los mecanismos necesarios para almacenar información en el LRS y para consultar la información contenida en el mismo. Por ejemplo, se pueden recuperar un Statement o un conjunto de Statements para realizar un reporte.
- **State API:** esta API es generalmente utilizada por los proveedores de actividades para llevar control del estado de dichas actividades. Por ejemplo, una persona edita una imagen como parte de una actividad. Esta imagen puede ser guardada en el LRS mediante esta API. Al final de la actividad, una aplicación puede consultar los

sucesivos cambios de estado de la imagen a lo largo del tiempo. También permite suspender, resumir, finalizar, etc., las actividades.

- **Activity Profile API:** permite almacenar datos relacionados a las actividades dentro del LRS. Esta información incluye una descripción completa de las actividades, sus definiciones, identificación, etc., para que el LRS pueda determinar correctamente a qué actividad se quiere acceder. Por ejemplo, como parte de una actividad, los usuarios juegan un juego y se guarda el puntaje obtenido mediante esta API.
- **Agent Profile API:** es similar a la State API y a la Activity API pero está enfocada en suministrar al LRS información respecto a los agentes. Es una manera de relacionar las diferentes formas en las que una persona ha sido identificada dentro del LRS asignando diferentes identificaciones y credenciales al agente.

3.3.3 Implementaciones de LRS

A la hora de implementar un LRS debemos decidir si el LRS formará parte del LMS o será implementado en su propio servidor [43]. Independientemente del tipo de LRS que se decida construir, hay una mínima funcionalidad que debe ser implementada, que es la API que vimos anteriormente .

3.3.3.1 LRS stand alone

En este caso tendremos un LRS totalmente autónomo respecto al/los LMS el cual, al implementar Tin Can, nos permitirá interactuar con una gran variedad de sistemas. Podemos ver al LRS como una caja negra que provee mecanismos de comunicación para enviar/consultar la información en él contenida sin interesarnos en la implementación interna del mismo.

Dicha implementación, la cual está representada en la figura 9, nos da independencia entre los LMS y el LRS; los LMS producirán contenido y será trackeado por el LRS mediante el uso de la API. Dicha API consta de un conjunto de servicios web, los cuales exponen toda la funcionalidad para la creación, actualización y consultas de actividades tincan. El LRS, no solo debe implementar el estándar Tin Can, sino a su vez las validaciones necesarias para garantizar la correctitud de los statements, y en caso de deberá informar el mensaje de error correspondiente.

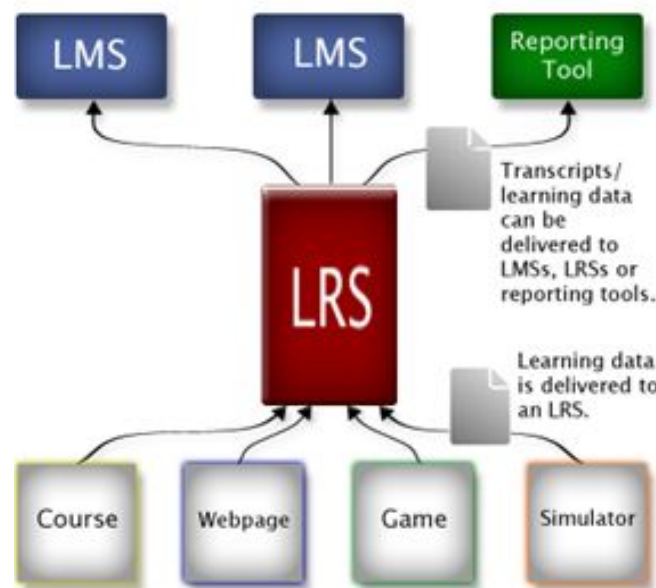


Figura 9: LRS independiente. Extraído de la documentación de Tin Can[33]

3.3.3.2 Integrando el LRS con el LMS

A continuación se mencionan algunas maneras de integrar un LRS con un LMS.

- **Ejecutando Tin Can e-learning:** tal vez la forma más sencilla de crear un LRS dentro de un LMS es utilizar Tin Can para lanzar actividades de aprendizaje. Los datos del alumno son tomados del LMS y los datos de las actividades son enviados a un LRS externo.
- **Enviar la información al LMS:** otra posibilidad de integración es enviar la información del LRS al LMS. Esta puede ser información de las actividades de aprendizaje ejecutadas desde el LMS u otras experiencias de aprendizaje.
- **Registrar otras actividades del LMS:** Tin Can puede registrar y seguir cualquier tipo de experiencias, no solo de e-learning. Moodle 2.7, por ejemplo, introduce herramientas para seguir cualquier actividad del usuario.
- **Embeber el LRS dentro del LMS:** una última opción de integración es poner el LRS dentro del LMS, como se muestra en la figura 10. Esta opción es más compleja y requiere mayor esfuerzo en su implementación que las anteriores. Con la gran cantidad de buenos LRS disponibles cabe preguntarse si es una opción fiable.

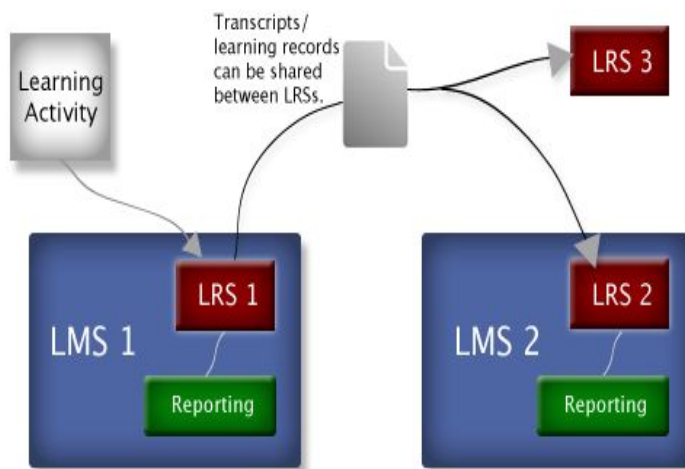


Figura 10: LRS como módulo del LMS. Extraído de la documentación de Tin Can[33]

3.3.4. Almacenamiento y gestión de datos

El LRS en cualquiera de las implementaciones vistas anteriormente tendrá que almacenar y consultar muchos más datos que un LMS tradicional. Dado que ahora almacenará todas las experiencias educativas de los alumnos, tanto las que se produzcan en el LMS como las que estemos siguiendo fuera del LMS.

Cabe destacar que las estructuras de datos de los LMS difieren entre sí y es por este motivo que Tin Can nos provee una gran versatilidad para representar una variedad de datos educativos, y a su vez poder persistirlos y consultarlos. De más está decir que la performance en la gestión de registros educativos cae exclusivamente en la propia implementación del LRS y en los recursos de hardware asignados al mismo.

Capítulo 4

4.1 Especificación del prototipo

Una vez analizadas las distintas implementaciones posibles de un LRS y las diferentes implementaciones existentes actualmente en el mercado, hemos decidido implementar nuestro propio LRS y su conjunto de APIs correspondientes siguiendo la especificación del estándar TinCan.

Además del LRS, desarrollaremos una serie de herramientas con el fin de integrar el mismo a la cotidianidad de la vida universitaria. Este conjunto de herramientas son:

- Una aplicación para celulares: utiliza la tecnología Android y permitirá a los alumnos interactuar con el LRS. Los alumnos podrán lanzar contenido que le resulte de interés para un curso dado.
- Una extensión local de Moodle: dado que todos los contenidos actuales se encuentran en el LMS (Moodle en este caso), es necesario el desarrollo de un módulo para Moodle, con el cual se podrán lanzar los contenidos del mismo al LRS.

Para lograr esto proponemos una arquitectura distribuida, la cual contará con un sistema central, el LRS, el cual será el encargado de administrar y gestionar las solicitudes, mediante servicios web, proveniente desde los distintos LMS o desde el cliente móvil. La figura 11 ilustra los distintos componentes que intervienen en el proceso.

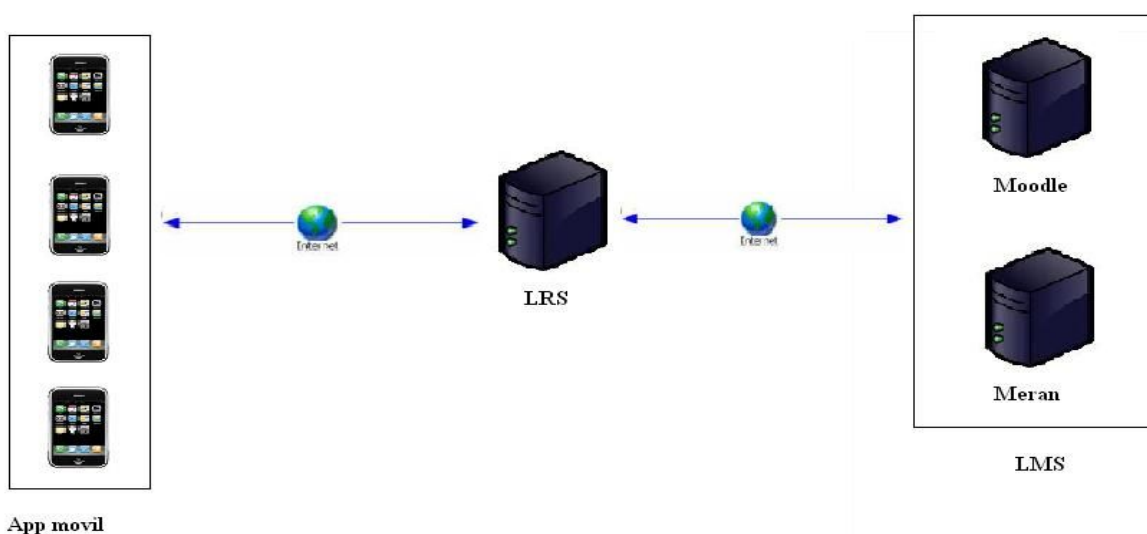


Figura 11: Esquema de la arquitectura del sistema.

Desde los distintos LMS, Moodle en nuestro ejemplo, los alumnos y profesores generan contenido. Un ejemplo es la publicación de un trabajo práctico por parte del profesor. Haciendo uso de la extensión desarrollada para Moodle, se registra en el LRS que el profesor publicó el trabajo práctico. Luego, desde la aplicación móvil, el alumno puede acceder al trabajo práctico.

Toda comunicación entre el LRS y los distintos clientes móviles y/o LMS se realizará utilizando servicios web, más específicamente servicios Rest. A su vez, el LRS tendrá la responsabilidad de validar la correctitud de los datos recibidos y de generar estadísticas a partir de dichos datos.

4.2 Prototipo del LRS

El objetivo de esta componente de software es realizar una implementación del estándar Tin Can, el cual sea capaz de interactuar con las demás componentes y almacenar las experiencias de los alumnos en una base de datos.

4.2.1 Requerimientos

Describiremos el prototipo a implementar a través de sus requerimientos. A éstos los distinguiremos entre requerimientos funcionales y no funcionales. También necesitaremos una distinción entre las funcionalidades del cliente móvil, y las funcionalidades propias del LRS.

Los requerimientos funcionales son los que definen el comportamiento interno del sistema: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran como los casos de uso serán llevados a la práctica. Los no funcionales especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Esta lista de requerimientos definirá luego los Casos de uso

Requerimientos funcionales del LRS

| ID | Nombre | Descripción |
|----------|--------------------|---|
| RF-LRS-1 | Autenticar | El sistema debe corroborar la identidad del Actor. |
| RF-LRS-2 | Administrar Grupos | El sistema debe permitir crear, modificar, eliminar y consultar grupos. |

| | | |
|-----------|------------------------|--|
| RF-LRS-3 | Administrar Agentes | El sistema debe permitir crear, modificar, eliminar y consultar agentes.. |
| RF-LRS-4 | Asignar agente a grupo | El sistema debe permitir agregar agentes a un grupo. |
| RF-LRS-5 | Crear statement | El sistema debe permitir crear un statement. |
| RF-LRS-6 | Recuperar statement | El sistema debe permitir recuperar un statement. |
| RF-LRS-7 | Generar reportes | El sistema debe permitir generar reportes. |
| RF-LRS-8 | Enviar notificaciones | El sistema debe enviar notificaciones a los usuarios móviles que correspondan. |
| RF-LRS-9 | Buscar statement | El sistema debe permitir buscar statements. |
| RF-LRS-10 | Buscar Actores | El sistema debe permitir buscar actor. |

Requerimientos no funcionales del LRS

| ID | Nombre | Descripción |
|-----------|---------------------|--|
| RNF-LRS-1 | Autenticar | El sistema debe corroborar la identidad del usuario. |
| RNF-LRS-2 | Validaciones | El sistema debe validar el ingreso de los datos. |
| RNF-LRS-3 | Tiempo de respuesta | El sistema debe responder de manera rápida a requerimientos tanto desde la administración Web como a requerimientos del cliente móvil. |

4.2.2 Casos de uso

El propósito de los casos de uso es describir en lenguaje natural la funcionalidad completa de un sistema a desarrollar y su empleo se realiza en el proceso de especificación de requisitos del sistema. Permiten capturar y definir los requisitos que debe cumplir una aplicación.

En este apartado utilizaremos los casos de uso como forma de acercar al lector las funcionalidades que la aplicación va a desempeñar frente al usuario. Sin embargo, no

vamos a profundizar demasiado en las posibilidades que ofrece esta técnica, sino que vamos a limitarnos en aquellos aspectos que son más ilustrativos y que ayuden a comprender mejor qué es lo que realiza la aplicación.

4.2.2.1 Diagrama de casos uso del LRS

Utilizaremos el diagrama de caso de usos que se muestra en la Figura 12, como medio para conceptualizar el alcance del sistema y poder identificar los actores que interactúan con el mismo (el LRS).

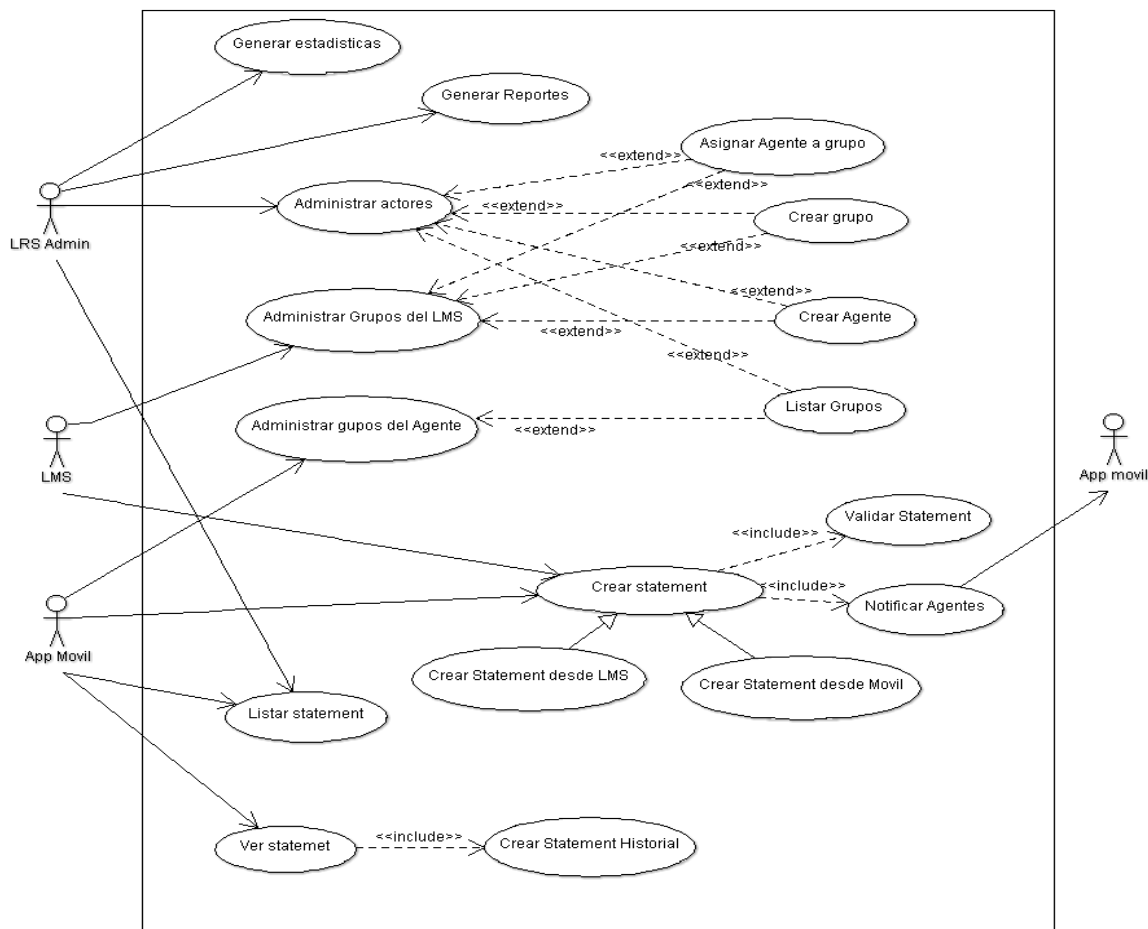


Figura 12: Diagrama de casos de usos

4.2.2.2 Descripción de los casos de uso del LRS

Describiremos los principales casos de uso que implementaremos para el desarrollo del prototipo, el resto de los casos de usos para el funcionamiento del LRS serán detallados en el anexo 1.

| Nombre | Listar statements |
|-----------------|--|
| Actor | Admin, Usuario móvil |
| Descripción | Se invoca al servicio para obtener obtener statements pasándole como parámetros los filtros de búsqueda correspondientes. |
| Precondiciones | El Actor se encuentra autenticado. |
| Flujo principal | 1- El actor correspondiente invoca al servicio, con alguno de los parámetros opcionales para la búsqueda, 2-El sistema retorna la lista de statements |

| Nombre | Crear statement |
|-------------------|--|
| Actor | LMS, usuario móvil |
| Descripción | Se crea un nuevo statement. |
| Precondiciones | El Actor se encuentra autenticado |
| Flujo principal | 1 - Se invoca al servicio crear statement. 2 - El sistema valida los datos enviados. 3 - El sistema crea un nuevo statement. |
| Flujo alternativo | 2.1 - Los datos son incorrectos, el sistema retorna el mensaje de error. |
| Postcondiciones | Existe un nuevo statement en el LRS. |

| Nombre | Validar Statement |
|---------------|---|
| Descripción | Se valida que el statement cumpla con el standard de TIn Can. Para esto se debe validar los campos obligatorios (Actor, Verbo, Objeto) y validar los campos opcionales que el mismo pueda tener. En caso de no pasar la validación, se debe informar |

| | |
|-------------------|--|
| | correctamente el o los atributos mal formados. |
| Flujo principal | <p>1- El sistema valida que exista el Actor.</p> <p>2- El sistema valida que exista el Verbo.</p> <p>3- El sistema valida el Objeto.</p> <p>4- El sistema valida el resultado. Opcional.</p> <p>5- El sistema valida el contexto(Opcional).</p> <p>6- El sistema valida el attachment (Opcional).</p> |
| Flujo alternativo | <p>1.1- El sistema retorna mensaje de error.</p> <p>2.1- El sistema retorna mensaje de error.</p> <p>3.1- El sistema retorna mensaje de error.</p> <p>4.1- El sistema retorna mensaje de error.</p> <p>5.1- El sistema retorna mensaje de error.</p> <p>6.1- El sistema retorna mensaje de error.</p> |

| Nombre | Listar Statements |
|-------------------|---|
| Actor | Admin, App móvil |
| Descripción | <p>Se listan los statements en orden cronológico decreciente.</p> <p>Se debe utilizar el usuario como filtro requerido y además se puede utilizar filtros opcionales como grupo, verbo, fecha, etc.</p> <p>Por cuestiones de performance se debe limitar a 20 el resultado, para esto se debe permitir paginación de los statements</p> |
| Precondiciones | El Actor se encuentra autenticado. |
| Flujo principal | <p>1-El admin o la app móvil invocan al servicio de búsqueda de statements, al mismo contienen parámetros opcionales, como grupo, rango de fechas, verbo.</p> <p>2-El sistema retorna la lista de statements.</p> |
| Flujo alternativo | 2.1-En caso de no encontrar statements, el sistema informa con el mensaje correspondiente. |

| Nombre | Ver Statement |
|-----------------|---|
| Actor | App móvil |
| Descripción | Se recupera un statement específico, junto a toda la información perteneciente al mismo, como lo es el contexto y los attachments. |
| Precondiciones | El actor se encuentra autenticado. |
| Flujo principal | <p>1-La app móvil invoca al servicio para ver un statement, con el id como parámetro requerido.</p> <p>2-El sistema retorna el statement completo. Esto incluye la información del contexto y los adjuntos.</p> |

4.2.4 Descripción de las clases principales del modelo

Describiremos brevemente las principales clases del modelo de datos que nos provee TinCan. Las mismas son:

- **Statement:** es la clase más significativa del modelo. Representa la construcción que consiste en <el actor (aprendiz)> <verbo> <objeto>, con <resultado>, en <contexto> para realizar un seguimiento de un aspecto de una experiencia de aprendizaje. Un conjunto de varios estados se puede utilizar para realizar un seguimiento más detalle de una experiencia de aprendizaje. Esta clase es la que representa “Yo hice esto”.
- **Actor:** representa a una persona o un grupo. Si pensamos un statement como una construcción “Yo hice esto”, representa el “Yo” dentro de la construcción.
- **Agent:** es la persona que interactúa con el LRS, por lo general los agentes interactúan con el LRS mediante el cliente móvil.
- **Group:** un grupo se compone de una colección de actores y/o de otros grupos.
- **Verb:** el verbo define la acción entre actor y objeto del statement. Representa el “hice” dentro del statement.
- **Activity:** una actividad es un tipo de objetos que representa el “esto” dentro del statement. Es algo con lo que el actor puede interactuar. Puede ser cualquier experiencia que se quiera registrar combinada con un verbo.
- **Context:** el campo de "contexto" ofrece un lugar para añadir un poco de información contextual para un statement. Es capaz de almacenar información como el instructor de una experiencia, si esta experiencia ocurrió como parte de un equipo de actividades, o cómo una experiencia encaja en alguna actividad más amplia.
- **Result:** un campo opcional que representa un resultado medido en relación con la statement en la que está incluido.
- **Object:** Actividad, Actor, u otro Statement que es el objeto de la statement. Representa el “esto” dentro del statement.

4.2.5 LRS interfaz pública

El LRS provee canales de comunicación para que cualquier sistema (LMS, app móvil, otro LRS, etc) pueda acceder a los statements ya sea para su consulta como para la creación de los mismos.

Para lograr esto el LRS expone una serie de servicios web los cuales están definidos en su especificación, a continuación se detallan aquellos que serán implementados para el funcionamiento del prototipo.

| Endpoint | Método | Descripción |
|---|--------|--|
| /actor/create | POST | Crea un actor. |
| /actor/get/{idActor} | GET | Retorna un actor con el id especificado. |
| /actor/addToGroup/{idGroup}/{idAgent} | GET | Agrega un agente a un grupo |
| /actor/getGroups/{idAgent} | GET | Retorna los grupos en los cuales un agente es miembro. |
| /verb/get/{idVerb} | GET | Retorna un verbo específico. |
| /verb/getAll | GET | Retorna todos los verbos. |
| /statement/create | POST | Se agrega el statement enviado al LRS |
| /statement/get/{idStatement} | GET | Retorna el statement con el el ID indicado |
| /statement/listStatements/{idActor}/{idGroup} | GET | Retorna la lista de statements para un actor y grupo dado. |
| /statement/getExtensions | GET | Retorna las extensiones para un statement. |
| /statement/addExtensions | POST | Agrega extensiones a un statement |

| | | |
|-------------------------------------|------|--|
| /statement/attachment/{idStatement} | GET | Retorna los archivos adjuntos para un statement. |
| /statement/attachment | POST | Agrega un archivo adjunto a un statement. |

4.3 Extensión para Moodle

Definido el LRS y la interfaz pública del mismo, el LRS está en condiciones que diversos sistemas puedan interactuar con él. Estos sistemas pueden ser LMS, aplicaciones móviles, otros LRS, etc.

Dado que para poner en marcha el prototipo necesitamos una fuente de datos lo más próximo a la realidad de la Facultad de Informática, por tanto decidimos utilizar uno de los LMS que se utilizan actualmente en dicha casa de estudio, una implementación de Moodle. Esta plataforma simulará la cátedra de “Laboratorio de Software”, la misma cuenta con alumnos, contenido y foros los cuales nos interesa lanzar al LRS como statements (experiencias educativas).

Con el fin de lograr esta tarea, será necesario crear una “extensión local” (plugin) para Moodle, la cual permitirá:

- Lanzar al LRS usuarios y cursos los cuales conformarán los agentes y los grupos del lado del LRS.
- Lanzar al LRS archivos, encuestas, foros, tareas, URL, etc que conformarán las actividades
- Registrar toda la actividad de los usuarios del LMS en el LRS.
- Registrar cambios en los contenidos del LMS.

4.4 Aplicación móvil

Los usuarios finales, en este caso los alumnos de la facultad, interactúan con el LRS por medio de una aplicación móvil la cual correrá en dispositivos con sistema operativo Android. Este desarrollo que hemos decidido llamar “LRSapp” será una herramienta que le permitirá interactuar con el LRS y acceder a sus cuentas en los LMS.

4.3.1 Alcance

El LRSapp es un desarrollo en el cual los alumnos podrán:

- Asociarse a un LMS dado.

- Consultar contenido publicado para un LMS asociado, ya sea que fuese producido por él mismo o bien por otro alumno.
- Responder (generar un sub statement) a una experiencia de aprendizaje dada.
- Generar una nueva experiencia de aprendizaje (generar un statement) para un grupo dado.

4.3.2 Requerimientos

Además de los requerimientos anteriormente mencionado para la construcción del LRS, el LRSapp añade una serie de requerimientos los cuales son propios de un desarrollo móvil. Estos requerimientos son los siguientes:

- Necesidad de un diseño amigable e intuitivo. Los usuarios de este tipo de aplicaciones están acostumbrado a un cierto tipo de interfaces gráficas que respetan una estética común.
- Fácil distribución de la misma. Dado que este desarrollo será utilizado por un número variado de alumnos, es necesario un mecanismo para distribuirla de manera eficiente.
- Capacidad de operar sin conexión a internet. Los dispositivos móviles suelen quedarse sin conexión internet, es por esta razón que debemos contar con un mecanismo para persistir los datos en el mismo.
- Manejo de errores. El LRSapp debe ser capaz de recuperarse ante errores inesperados.

Capítulo 5

5.1 El LRS

Como se mencionó en capítulos anteriores, uno de los componentes principales del sistema, si no el más importante, es el LRS. Para la implementación del mismo se analizaron distintas opciones de uso libre. Estas implementaciones incluyen los LRS alojados en servidores de terceros (en la nube) o distribuciones desplegadas en nuestros propios servidores.

Luego de analizar ventajas y desventajas de las distintas opciones encontradas decidimos implementar nuestro propio LRS. El mismo cuenta con la funcionalidad mínima requerida para el desarrollo de nuestra tesina y fue desarrollado siguiendo las especificaciones provistas por Tin Can API.

El LRS está conformado por una colección de APIs bien definidas, cada una enfocada en satisfacer distintas necesidades. La más importante, relacionada a la transferencia de statements desde y hacia el LRS, es la Statement API. Es el mecanismo básico de comunicación de Tin Can API.

Por ser la más importante y vital del LRS, nuestro desarrollo comprende la implementación de dicha API. Para su desarrollo se optó por realizar una aplicación web que consta de dos partes:

- El primer desarrollo consta de la implementación del LRS, esto incluye modelo de dominio, capa de persistencia, capa de negocio e interfaz pública. Este desarrollo es el corazón de nuestro prototipo.
- El segundo desarrollo es de un sencillo utilitario para poder administrar el LRS mediante un navegador web.

5.1.1 Herramientas para el desarrollo

Dadas las características ya estudiadas del prototipo a implementar, al seleccionar las herramientas nos hemos basado en dos aspectos, el primero de ellos que sean herramientas con alta trayectoria en el mercado, esta característica nos garantiza que las mismas se encuentran en un nivel de madurez. El segundo aspecto es el conocimiento que los autores de esta tesina tienen sobre las mismas. Por las razones ya citadas hemos elegido Java como lenguaje de programación y utilizaremos el framework Spring [44] para facilitar y agilizar el desarrollo del prototipo. Para la persistencia de los datos utilizaremos el

motor de bases de datos MySQL [45] y como servidor de aplicaciones utilizaremos Tomcat [46].

Todas las herramientas mencionadas, son de uso libres y han sido estudiadas en distintas asignaturas durante la carrera.

5.1.2 Implementación del modelo

El primer paso del desarrollo del LRS, es la implementación del modelo de datos, el cual será transversal a las capas del sistema. Abordaremos específicamente el tipo de dato “Statement” por ser la unidad de aprendizaje. Por esto describiremos los principales atributos que contiene un statement junto a sus principales objetos asociados.

| Propiedad | Descripción | Requerido |
|-----------|--|-----------|
| id | UUID asignado por el LRS . | Si |
| actor | Propietario del statement, puede ser Agente o Grupo. | Si |
| verb | Acción tomada por el Actor | Si |
| object | Puede ser una Actividad, Agente, u otro Statement. | Si |
| context | El contexto provee información que enriquece al statement | No |
| timestamp | Fecha y hora de creación del statement. | No |
| stored | Fecha y hora de registros del statement en el LRS. Es seteado por el LRS | Si |

| | | |
|-------------|---|----|
| attachments | Se adjunta datos al statement para enriquecerlos. | No |
|-------------|---|----|

De esta serie de atributos, describiremos los principales (los requeridos, para que un statement sea válido).

El Actor define quien realizó la acción del statement. El actor puede ser un Agente o un Grupo. Cuando el Actor es un agente, este objeto identifica o bien a una persona o a un sistema. En nuestro caso particular los agentes son por un lado los alumnos, y por otro lado el LMS. El otro caso posibles es que el actor sea un grupo. Un Grupo representa una colección de agentes y puede ser utilizado en la mayoría de las situaciones que un agente puede ser utilizado.

El Verbo define la acción entre un Actor y una Actividad. El Verbo en un statement describe la acción realizada durante la experiencia de aprendizaje. El standart Tin Can no especifica ningún verbo en particular. Es por esto que los mismos serán definidos por los administradores del LRS, en función al dominio que se esté implementado el mismo.

El Objeto define la cosa sobre la que se actuó. El objeto de una statement puede ser una actividad, un agente o grupo, un statement. Los objetos deben incluir una propiedad "objectType", la cual nos indica el tipo de objeto. Si no se especifica esta propiedad, el objeto es de tipo "Activity". Otros valores válidos son: "Agent", "Group", "SubStatement" o "StatementRef".

5.1.3 Implementación de la capa de negocio

La implementación del LRS requiere la correctitud de los statements, para luego poder procesarlos y finalmente persistirlos en la base de datos. Para lograr esto seguimos la especificación del estándar e implementamos una serie de clases para cumplir con el objetivos de esta capa. La validación de los statements en xAPI solo se realiza sobre su sintaxis, no sobre la semántica de sus datos. Tin Can impone como requerimiento para el LRS validar la estructura de los statements. En nuestro desarrollo se verifica que los verbos, actividades, actores, etc. se encuentren correctamente definidos.

5.1.4 Statement API

La especificación Tin Can establece que esta API debe manejar la transferencia de statements a través de distintos métodos HTTP RESTful:

- PUT Statements: almacena statements con un identificador específico.

- **POST Statements:** almacena uno o más statements. A diferencia del método PUT, que especifica un único statement, este método puede utilizarse para almacenar múltiples statements o almacenar uno sin necesidad de generar previamente su identificador. Retorna los identificadores de los statements almacenados.
- **GET Statements:** permite recuperar uno o varios statements almacenados en el LRS. Puede recuperarse un statement en particular especificando su identificador o un grupo de statements mediante distintos filtros (fecha, agente, actividad, etc.).

Todos los métodos HTTP RESTful de la Statement API son invocados direccionando las peticiones hacia **http://endpoint/statements**, donde **http://endpoint** es la URL de nuestro LRS.

5.1.4.1 Implementando la Statement API

Para la implementación de LRS, hemos optado por utilizar Java como lenguaje de programación y el framework Spring. La elección de estas herramientas es debido a la experiencia que tenemos sobre las mismas.

PUT Statements: Es la manera más simple de almacenar un statement. Requiere un sólo parámetro, “**statementId**” y es el identificador del statement que se quiere almacenar.

URL: **http://endpoint/statements?statementId=xxxxxxx**

El LRS responde con el código de estado correspondiente, por ejemplo “204 No Content” en caso de procesarse correctamente el requerimiento.

POST Statements: No difiere mucho del método anterior. Al igual que el PUT, sirve para almacenar statements en el LRS pero de manera masiva. En este caso no es necesario especificar los identificadores de los statements, dejando al LRS generarlos en caso de ser necesario. Este método no requiere parámetros y responde retornando el código de estado correspondiente y el identificador de cada uno de los statements almacenados.

URL: **http://endpoint/statements**

GET Statements: Recupera uno o varios statements del LRS. Los parámetros son opcionales, variando según los criterios de búsqueda. Se puede especificar un identificador de statement en particular, los statements almacenados a partir de una fecha, los statements relacionados a un agente en particular, etc. Retorna el código de estado correspondiente y la lista de statements que satisfacen los criterios especificados en los parámetros.

URL: <http://endpoint/statements>

Ejemplos:

- <http://endpoint/statements?statementId=xxxxxx>: Devuelve el statement con el id xxxxxx
- <http://endpoint/statements?agent={mbox:'mailto:agente@lrs.com', type: 'Agent'}>: Devuelve los statements generados por el agente cuyo mail es 'agente@lrs.com'.
- <http://endpoint/statements?since=xxxxxx>: Devuelve los statements almacenados a partir de la fecha xxxxxx

5.2 Desarrollo del plugin para moodle

Para el desarrollo del plugin para la plataforma Moodle se optó por implementar una extensión local. Dicha extensión consiste en un archivo con formato .zip que contiene toda la información necesaria para su instalación y correcto funcionamiento. A continuación se explica en qué consiste una extensión Moodle y cómo se debe proceder para instalarla.

La manera recomendada para añadir funcionalidad al Moodle es crear una nueva extensión. Las extensiones locales son las más convenientes para agregar funcionalidades que escapan a las extensiones estándares de Moodle.

5.2.1 Extensiones locales personalizadas

Las extensiones locales son usadas en los casos donde las extensiones estándares no se aplican a la funcionalidad que se desea agregar. Ejemplos de tales casos, entre otros, son los siguientes:

- Componentes que consumen y manejan eventos y se comunican con sistemas externos
- Definiciones personalizadas de servicios web y funciones externas.
- Aplicaciones que extienden Moodle a nivel sistema (hub server, amos server, etc.)
- Configuraciones de usuario y administración personalizadas
- Extensiones del bloque de navegación con menús personalizados

5.2.2 Características de las extensiones

La extensión consiste en una serie de archivos .php estructurados como se muestra en a continuación:

- nombre_extension
 - db

- o access.php
- o events.php
- o install.php
- o install.xml
- o uninstall.php
- o upgrade.php
- lang
 - en
 - o local_tesina.php
- o default.php
- o lib.php
- o settings.php
- o version.php

En la carpeta “db” se encuentran los archivos de configuración correspondientes a la base de datos relacionada a la extensión.

install.xml: se ejecuta durante la instalación en caso de encontrarse una nueva versión de la extensión. Contiene la estructura de las tablas relacionadas a la extensión.

install.php: se ejecuta justo después de ejecutarse el archivo install.xml. Se puede utilizar, por ejemplo, para inicializar las tablas relacionadas con la extensión e insertar los datos básicos para su funcionamiento.

uninstall.php: se ejecuta durante la desinstalación de la extensión. Por ejemplo, en el se puede hacer backup de las tablas relacionadas a la extensión.

upgrade.php: se ejecuta para actualizar la extensión. Esto se produce cuando el archivo versión.php se modifica.

access.php: en este archivo se definen las capacidades de la extensión. Si la extensión puede leer, modificar, eliminar, etc. los datos de la base de datos, es en este archivo donde se configura.

events.php: en este archivo se definen los manejadores de los eventos que se quieren manejar a través de la extensión. Un ejemplo de evento es la creación de un usuario. Si se desea realizar cualquier tipo de acción luego de crearse un usuario, en este archivo se define un “handler” para dicho evento y se indica qué función se debe ejecutar para manejar el evento.

La carpeta “lang” contiene los archivos de recursos para los distintos idiomas

soportados por la extensión. Cada idioma posee una carpeta dentro de la carpeta “lang”. Por ejemplo, para el idioma Inglés, se debe crear una carpeta con el nombre “en”. Dentro de cada carpeta de idioma se encuentra el archivo “local_nombreextension.php”.

local_nombreextension.php: archivo de recursos.

Por último, dentro de la carpeta raíz de la extensión (“nombre_extension”), se definen los siguientes archivos:

version.php: indica la versión del script. Se debe incrementar con los cambios para indicar a Moodle que debe actualizar la extensión.

lib.php: librería de funciones. Es incluida automáticamente por Moodle. Aquí se definen todas las funciones correspondientes a la extensión.

settings.php: Opciones de configuración. Las opciones de configuración aquí definidas se pueden acceder en el menú de administración de la extensión dentro del menú de administración de extensiones locales del Moodle. Luego, desde las funciones definidas en “lib.php” podemos acceder a dicha configuración.

Instalación de la extensión

La instalación de la extensión se realiza de manera sencilla. El contenido del archivo .zip que contiene la extensión debe descomprimirse dentro de la carpeta “/local” (“\local” en sistemas Windows) situada dentro de la estructura de directorios del sistema Moodle. A continuación accedemos como usuario administrador al Moodle. Moodle detecta la nueva extensión y nos presentará la pantalla de la figura 14.

Comprobación de 'plugins'

Esta página muestra las extensiones (plugins) que pueden requerir su atención durante la actualización. Los elementos resaltados incluyen nuevas extensiones (plugins) que están a punto de ser instalados, los que van a ser actualizados y las extensiones anteriores que ahora faltan. Los módulos externos (add-ons) también se destacan. Se recomienda que compruebe si hay versiones más recientes de los módulos externos disponibles y actualice su código fuente antes de continuar con esta actualización de Moodle.

Compruebe actualizaciones disponibles

Última comprobación realizada el 30 de noviembre de 2014, 19:45

Número de extensiones (plugins) que requieren atención durante esta actualización: 1

[Mostrar la lista completa de extensiones \(plugins\) instalados](#)

| Nombre de la extensión | Directorio | Origen | Versión actual | Nueva versión | Requiere | Estado |
|--------------------------------------|---------------|----------------|----------------|---------------|-------------------|-----------------|
| Extensiones (plugins) locales | | | | | | |
| Tesina | /local/tesina | Módulo externo | | 2014101501 | Moodle 2010112400 | Para instalarse |

Recargar

Actualizar base de datos Moodle ahora

[Purgar todas las cachés](#)

Figura 14: Instalación de extensión en Moodle

En esta pantalla Moodle lista todas las extensiones que requieren, o bien ser instaladas, o necesitan actualizarse. En nuestro caso nos muestra el nombre de la extensión, su ubicación, el tipo de extensión, el número de versión y su estado. Para proceder con la instalación damos click al botón “Actualizar base de datos de Moodle ahora”.

A continuación se mostrará otra pantalla indicando que la instalación se realizó correctamente o los errores detectados durante la misma.

5.2.3 Configuración de la extensión

En caso de haber definido opciones de configuración, en el archivo settings.php, visto anteriormente, podremos configurar dichas opciones mediante el panel de administración del Moodle. Para ello ingresamos al Moodle con una cuenta de usuario con perfil de administrador. A la izquierda de la pantalla veremos el panel de “Administración”. Dentro del panel debemos seleccionar la opción “Administración del Sitio” > “Extensiones” > “Extensiones Locales”. Dentro de “Extensiones Locales” encontraremos la opción para gestionar las extensiones locales y el menú que hemos definido para nuestra extensión, en este ejemplo “Configurar Tesina”, como se muestra en la figura 15.

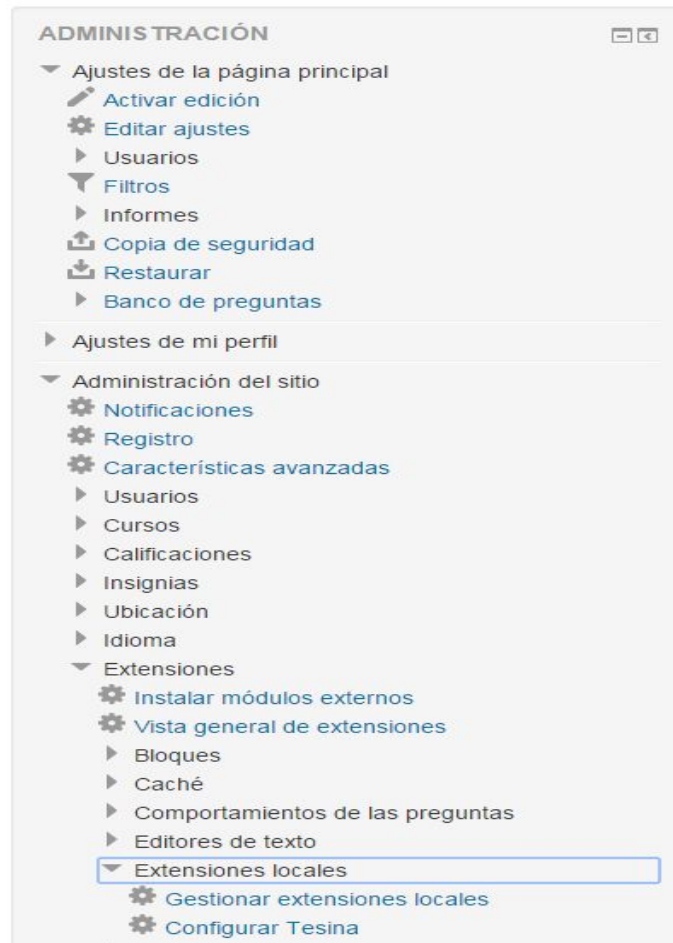


Figura 15: Configuración de la extensión.

Si accedemos a “Configurar Tesina”, Moodle mostrará las opciones de configuración que hayamos definido cuando creamos la extensión. Esto se aprecia en la figura 16.

Configurar Tesina

URL Endpoint Valor por defecto: http://localhost/prueba
endpoint

URL correspondiente al endpoint

Figura 16: Configuración de la extensión

Tanto el título del menú (“Configurar Tesina) como el nombre de las opciones (“URL

Endpoint”), sus descripciones y valores por defecto se definen en el archivo settings.php.

Funcionamiento

En las secciones anteriores se describió brevemente en qué consiste y cómo se instala una extensión para Moodle. En esta sección se explicará cómo funciona la extensión por nosotros desarrollada como parte integral de la tesis propuesta.

El objetivo de nuestra extensión es registrar ciertas actividades realizadas por los usuarios de la plataforma Moodle en el LRS comprendido por la tesis. El tipo y cantidad de actividades comprendidas en la plataforma Moodle son muchos y variadas. Sin embargo, la extensión está enfocada en un grupo reducido de las mismas, siendo estas actividades las que presentan mayor interés a nuestro entender. Dichas actividades son las siguientes:

- Archivos
 - Crear
 - Borrar
 - Descargar
- Curso
 - Crear
 - Eliminar
 - Agregar usuario a curso
 - Eliminar Usuario del curso
- Encuestas
 - Crear
 - Responder
- URL
 - Crear
 - Borrar
 - Ver
- Foros
 - Crear
 - Borrar
 - Leer
 - Comentar
- Tareas

- Crear
- Ver
- Completar
- Usuario
 - Crear
 - Eliminar

Nuestro objetivo es registrar la realización de estas actividades por parte de los usuarios de Moodle. Es aquí donde entra en juego nuestra extensión. Uno de los archivos fundamentales para el funcionamiento de la extensión es el archivo “events.php”, el cual se encuentra dentro de la carpeta “db”. Es en este archivo donde le indicamos a Moodle que, ante un evento determinado, le indique a nuestra extensión que el evento se produjo. Es ahora nuestra responsabilidad determinar qué hacer con dicho evento. En nuestro caso, registrar la actividad en el LRS. Ejemplo:

```
'evento' => array (
    'handlerfile' => 'nombre_archivo',
    'handlerfunction' => 'nombre_funcion',
    'schedule' => 'instant',
    'internal' => 1)
```

En el ejemplo hemos definido un manejador para el evento '**evento**'. El manejador se define mediante un arreglo que contiene la siguiente información:

- **handlerfile**: en que archivo se encuentra la función que maneja el evento.
- **handlerfunction**: nombre de la función, dentro del handlerfile, responsable de manejar el evento.
- **schedule**: indica el momento en el cual Moodle delega el evento a nuestra extensión. En el caso de **instant**, se delega al momento de producirse el evento.
- **internal**: Indica si la función que maneja el evento debe ser re invocada en caso que su ejecución no resulte exitosa.

Cuando el Moodle detecta que alguno de los eventos que nos interesa se produce, busca en nuestra extensión si hemos establecido un manejador para el mismo. Si lo encuentra, busca el **handlerfile** e invoca la **handlerfunction**. La función definida como handlerfunction recibe como parámetro el evento producido y debe retornar true o false

indicando éxito o fracaso en su ejecución. Ejemplo:

```
function handlerfunction($evento)
{
    OPERACIONES DE LA EXTENSION
    return true;
}
```

En el cuerpo de la función realizamos las operaciones correspondientes, en el caso nuestro comunicarnos con el LRS y registrar la actividad.

5.3 Desarrollo de la aplicación móvil

El cliente móvil es la componente del sistema que más interactúa con los usuarios, es por esta razón que este desarrollo intenta abarcar la mayor cantidad de dispositivos posibles. Teniendo en consideración este objetivo pero sin dejar de lado que se desarrolla un prototipo hemos decidido limitar la variedad de dispositivos a aquellos que corren bajo el sistema operativo Android [47], a su vez hemos limitado aún más el conjunto de los mismos para aquellos que tengan la versión de Android 4.4.2 o superior. Aun así el conjunto de dispositivos es sumamente grande y presenta un gran desafío lograr que en los mismos la aplicación funcione de manera correcta. Otra ventajas al elegir la plataforma Android para el desarrollo es la capacidad de fácil distribución de la misma mediante la plataforma Google Play Store[48], la cual nos facilita el deploy de las aplicaciones en los clientes.

Una vez, resuelto el tema de las plataforma y la distribución del LRS, nos concentramos en desarrollo. Para abordar esta problemática hemos cambiado nuestro enfoque de una solución en Android nativo hacia un desarrollo híbrido [49]. Este tipo de desarrollo se está imponiendo en el mercado como la alternativa más viable al desarrollar aplicaciones móviles.

5.3.1 Aplicaciones híbridas

Una aplicación híbrida para móviles es quela la cual en su interior contiene un navegador web del dispositivo. Para su desarrollo se utilizan frameworks de desarrollo basados en lenguajes de programación web (HTML, CSS y JS). Además este tipo de aplicación contiene un desarrollo nativo el cual interactúa con las componentes propias del teléfono.

Los precursores de este tipo de aplicaciones es el proyecto Apache Cordova [50], la idea de este proyecto es basar gran parte de desarrollo en una componente WebView, la cual nos permite utilizar tecnologías web (HTML, CSS y JS) para evitar el desarrollo nativo, estas aplicaciones se ejecutan dentro de un contenedor por cada plataforma y nos provee una API para acceder a sensores, datos, estado de la red, etc. En la figura 17 vemos la arquitectura de una aplicación Apache cordova.

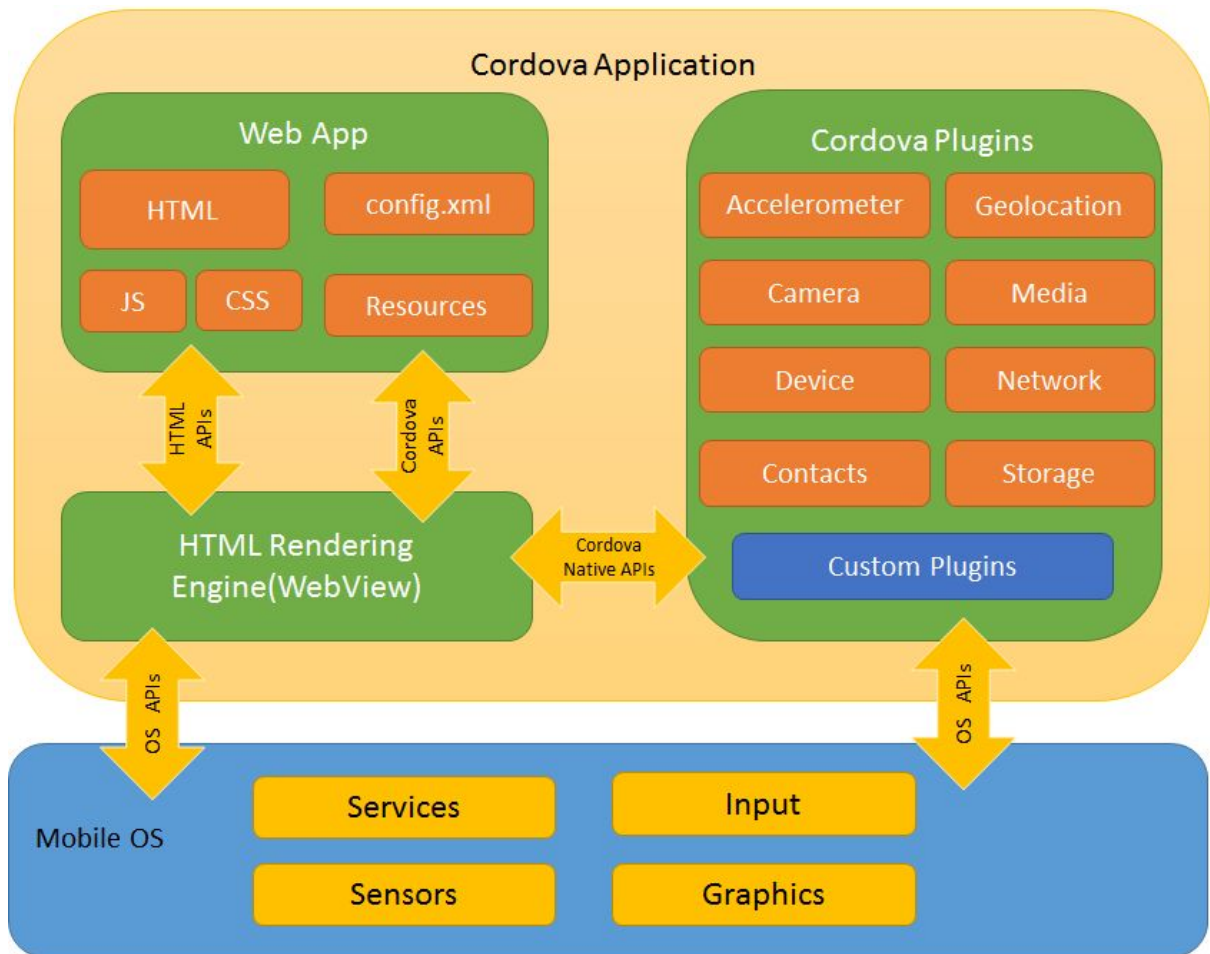


Figura 17: Arquitectura Cordova. Extraído de la documentación de Cordova[50]

Para nuestro desarrollo móvil hemos utilizado el framework Ionic [51] [52], al cual consideramos uno de los frameworks de aplicaciones híbridas más completos que actualmente hay en el mercado. Este framework combina el proyecto Cordova con el frameworks JavaScript Angular [53] y además provee un conjunto de soluciones de interfaces de usuario y una basta comunidad de desarrolladores. Es por esto que nuestra LRSapp fue construido utilizando estas tecnologías de vanguardia.

5.3.2 Estructura del LRSapp

Detallaremos las componentes principales del desarrollo del LRSapp utilizando Ionics como framework de desarrollo. Una vez configurado el framework y creado el proyecto nos encontraremos con la estructura de carpetas que todo proyecto Ionic contiene [54]. La misma se visualiza en la figura 18.



Figura 18: Estructura de archivos del LRSapp

De esta serie de archivos y carpetas destacaremos en primer lugar el archivo “config.xml”, del cual muchos aspectos del comportamiento de una aplicación pueden controlarse con este archivo de configuración global. Este archivo XML independiente de la plataforma declara basado en la especificación del W3C Empaquetado aplicaciones Web (Widgets) [55] y su propósito es especificar funciones de la API de Cordova centrales tales como plugins y configuración específica de la plataforma.

En segundo lugar nos interesa hacer foco en la carpeta “WWW”, la cual es donde pondremos el código de nuestra aplicación. La figura 19 nos muestra la misma junto a sus subcarpetas y archivos.



Figura 19: Carpeta WWW juntoa sus archivos y subcarpetas.

El archivo principal que se abrirá al cargar la aplicación se llama “index.html”. Aquí tendremos que cargar todo lo necesario y mostrar el contenido de la primera pantalla.

La carpeta “templates” está preparada para almacenar las plantillas o vistas de la aplicación (en algunas versiones no se crea por defecto). Aquí depositamos nuestros archivos HTML, los cuales conformarán las distintas vistas de nuestra aplicación. En la figura 20 podemos apreciar los mismos.

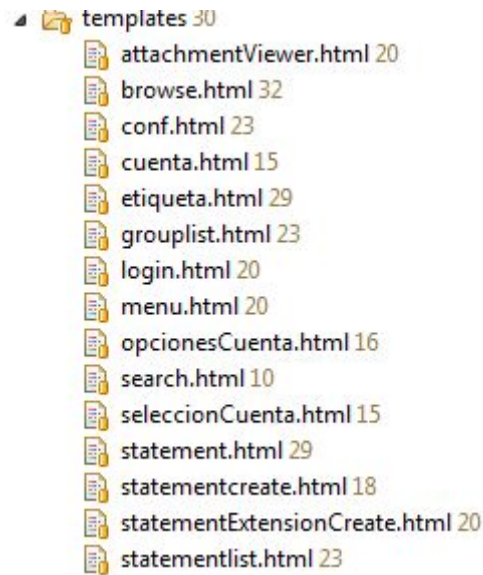


Figura 20: Carpeta con los archivos templates.

Nuestra carpeta “js” contendrá los archivos javascript necesarios para que nuestra aplicación funcione. La figura 21 muestra los archivos que contiene esta carpeta. De estos destacamos:

- app.js: En este fichero se inicializa el módulo de Angular que gestionará la aplicación.
- controllers.js: este archivo implementamos los controladores para las vistas.
- localStorage.js: en este archivo implementamos la persistencia de datos en el dispositivo.
- service.js: En este archivo se implementó todo lo referente a la comunicación vía servicios REST con el LRS.

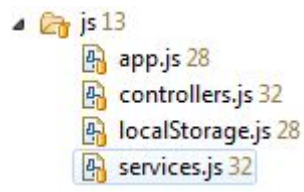


Figura 21: Carpeta con archivos JavaScript

Capítulo 6

6.1 Experiencias de usuarios

Los alumnos de la facultad, los que consideramos nuestros usuarios finales, interactúan con el LRS mediante el uso del LRSapp en su dispositivo móvil. Es por esto que pasamos a describir LRSapp y su funcionamiento. Como ya se mencionó anteriormente, no contamos con el apoyo de un diseñador gráfico, por eso hemos utilizados las componentes gráficas que nos provee el framework Ionics.

6.1.1 Interfaz

Hemos optado por utilizar las componentes de interfaz que nos provee el framework, para esto definimos una cabecera comun a todas nuestras vistas, la misma la podemos ver en la figura 22 y pasaremos a detallar a continuación.

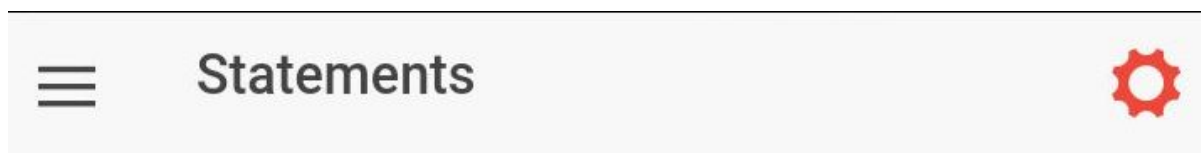





Figura 22: Cabecera común a todas las vistas del LRSapp

A la derecha del botón de retroceso se encuentra el boton de menu  , y como lo indica su nombre sirve para acceder al menú de la aplicación. Al presionarlo se desplegará el menú sobre el lado izquierdo de la pantalla.

A la derecha del botón de menú se muestra el título de la página actual, “**Configuración**” en la imagen del ejemplo.

Finalmente, sobre el margen derecho se encuentra el botón de opciones  , que, al presionarlo, nos mostrará las distintas opciones disponibles según la página en la que nos encontremos. Debajo de la sección mencionada se encuentra el contenido, en donde se muestran las distintas páginas de la aplicación.

En el margen superior, sobre la izquierda se encuentra el botón de retroceso  , la cual solo aparecerá cuando no se encuentre posicionado en el inicio de la app. Presionando este botón podemos volver a la página anterior a la página actual.

6.1.2 Menú principal

Para acceder al menú de la aplicación se debe presionar el botón de menú, descrito en el punto anterior. Al presionarlo se desplazará el menú sobre la parte izquierda de la pantalla. Presionar nuevamente el botón hará que el menú se oculte.

Otra manera de mostrar el menú es deslizando la pantalla hacia la derecha. Esto tendrá el mismo efecto que apretar el botón. Lo mismo se aplica para ocultarlo. Podemos hacerlo desplazando la pantalla hacia la izquierda. En la figura 23 podemos ver el menú desplegado.

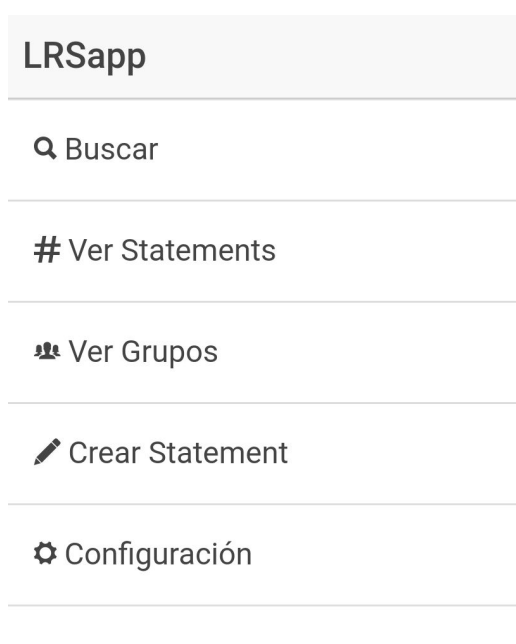


Figura 23: Menú principal del LRSapp

6.1.3 Configuración inicial

La configuración de la aplicación es bastante sencilla. Lo único que tenemos que configurar es la URL del LRS al cual queremos conectarnos. Para acceder a la configuración tenemos que apretar el botón de menú y seleccionar la opción **“Configuración”**.


La primera vez que iniciemos la aplicación, automáticamente se nos abrirá el menú de configuración. En la parte superior de la pantalla, bajo el título **“Configuración General”** es donde se configura la URL mencionada. Para ello solo tenemos que completar el campo **“URL LRS”** con la URL deseada. En el campo **“Statements por página”** tenemos la posibilidad de especificar cuántos statements queremos que se nos muestre en cada página

como resultado de las búsquedas. La figura 24 muestra la sección de configuración de la aplicación.



Figura 24 - Configuración del LRSapp.

6.1.4 Cuentas

Luego de realizar la configuración inicial, el siguiente paso es configurar las cuentas que queremos asociar a nuestra aplicación. Para ello debemos presionar el botón de menú y una vez desplegado el mismo seleccionar la opción “**Configuración**”. Para agregar una nueva cuenta, solo tenemos que presionar el botón  y se mostrará el formulario para completar los datos de la misma.

Nueva cuenta ✕

Descripcion

Homepage

Name

Password


 **Agregar**

Figura 25: Configuración de cuentas.

A continuación se detallan los campos de la cuenta, los mismos pueden visualizarse en la figura 25:

- **Descripción:** breve descripción de la cuenta. Sirve para que el usuario pueda distinguirla fácilmente en caso de configurar múltiples cuentas. Un ejemplo podría ser **“Moodle Info UNLP”**.
- **Homepage:** en este campo debe especificarse la URL de la cuenta deseada. En nuestro caso, la URL debe corresponderse a una cuenta de Moodle con la extensión de nuestra aplicación instalada.
- **Name:** nombre de usuario de la cuenta deseada. Debe corresponderse exactamente al nombre de usuario que se utiliza para acceder a la cuenta. En nuestro caso, el nombre de usuario del Moodle.
- **Password:** al igual que el nombre de usuario, corresponde a la contraseña utilizada para acceder a la cuenta. **Nota:** en la versión de la aplicación utilizada para esta tesis, la contraseña no es autenticada por lo tanto el contenido de este campo no es relevante.

Debajo del título **“Cuentas”** se listan las cuentas agregadas (figura 26).

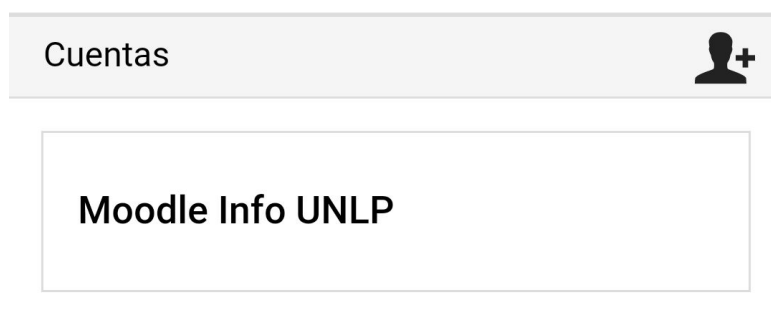



Figura 26: Listado de cuentas

Para eliminar una cuenta solo hay que presionar sobre la cuenta que se desea eliminar y se abrirá una ventana con la opción para eliminarla (figura 27).



Figura 27: Eliminar cuenta

Seleccionar cuenta

La aplicación permite configurar múltiples cuentas. Por ejemplo, podríamos dar de alta, además de la cuenta de ejemplo del Moodle, una cuenta relacionada con algún otro LMS. Sin embargo, siempre estaremos trabajando sobre una cuenta específica al momento de crear un statement, realizar una búsqueda o simplemente listando los statements. Para cambiar de cuenta y seleccionar la cuenta sobre la que queremos trabajar hay que presionar el botón de opciones  que se encuentra sobre el margen superior derecho de la pantalla. Al presionarlo se abrirá una ventana listando todas las cuentas configuradas en la aplicación.

Cuentas disponibles



| | |
|------------------|---|
| Moodle Info UNLP | ✓ |
|------------------|---|

| |
|-------------|
| Otra cuenta |
|-------------|

Figura 28: Seleccionar cuenta

La cuenta actual, es decir sobre la que se está trabajando actualmente, se indica con un tilde junto a su descripción (figura 28). Si se quiere seleccionar una distinta hay que presionar sobre la cuenta deseada. Una vez seleccionada, los statements listados, así como los nuevos statements estarán relacionados con la nueva cuenta.

6.1.5 Listar Statements

Al seleccionar esta opción de menú se accede al listado de statements. Los statements se listan ordenados por fecha de creación, de manera decreciente. De cada statement se muestra la fecha, el usuario, el curso, en caso de estar asociado a alguno, y el nombre o asunto del statement. Se utilizó un formato de tarjeta para representar cada uno de los statements, para que sea más amigable a los usuarios. La lista de statements puede apreciarse en la figura 29.

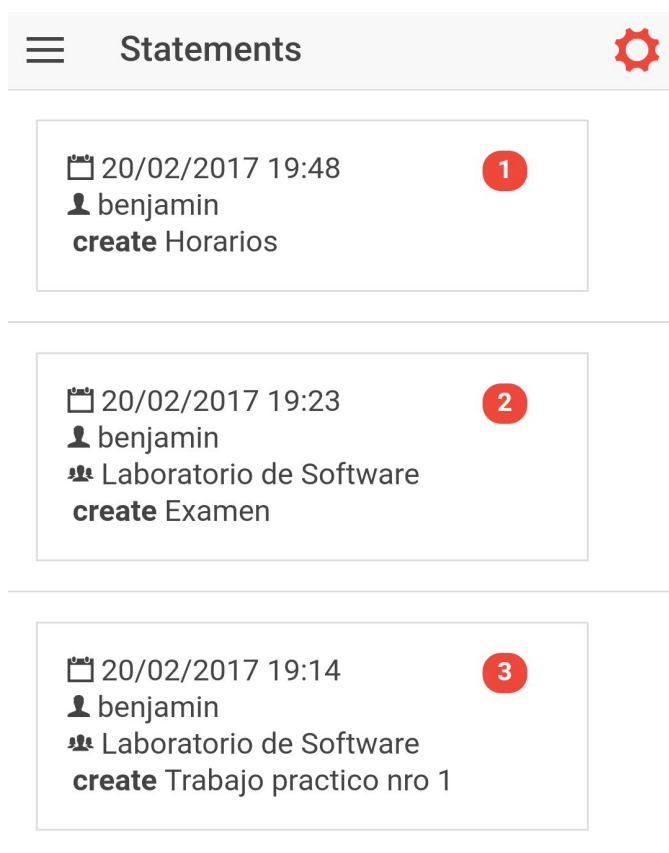


Figura 29: Listado de statements.

Presionando sobre algún elemento de la lista de statements se accede a la vista que nos mostrará el statement seleccionado junto al conjunto de datos que lo componen, esto puede verse en la Figura 30. Pasamos a describir las principales características de esta funcionalidad

Identificador del statement. Es generado por el LRS y es único para cada statement.

🏷️ Etiquetas del statement. Presionando sobre este campo se pueden agregar/eliminar las etiquetas del statement.


📅 Fecha de creación del statement.

👥 Grupo o curso relacionado al statement. Indica que la acción del statement está ligada a este grupo. En nuestro ejemplo sería la cátedra.

👤 Agente o usuario que realizó la acción del statement. En nuestro ejemplo sería el alumno.

Debajo del nombre del agente se muestra la acción realizada en el statement y la actividad relacionada a esa acción. En este caso se creó la actividad “Examen”.

En la sección “Adjuntos” se listan, en caso de haberlos, los adjuntos del statement.

Finalmente, en la sección “Substatements” se listan los substatements generados a partir del statement. Podemos pensar a los substatements como las respuestas generadas a partir del statement. En el ejemplo, algún alumno o profesor podría generar un substatement para responder a la pregunta planteada por el alumno. Para generar un substatement basta con presionar el .

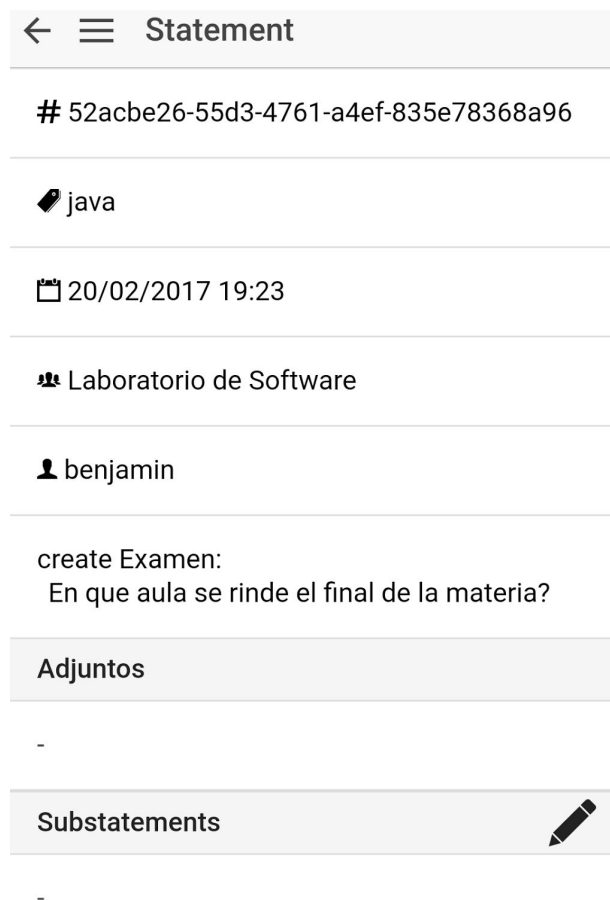


Figura 30: Detalle del statement

6.1.6 Buscar Statements

Esta opción de menú nos permite buscar statements utilizando una serie de filtros que se describirán a continuación y puede visualizarse en la figura 31.

☰ **Buscar** ⚙️

Grupo

Fecha desde

Fecha hasta

Etiquetas


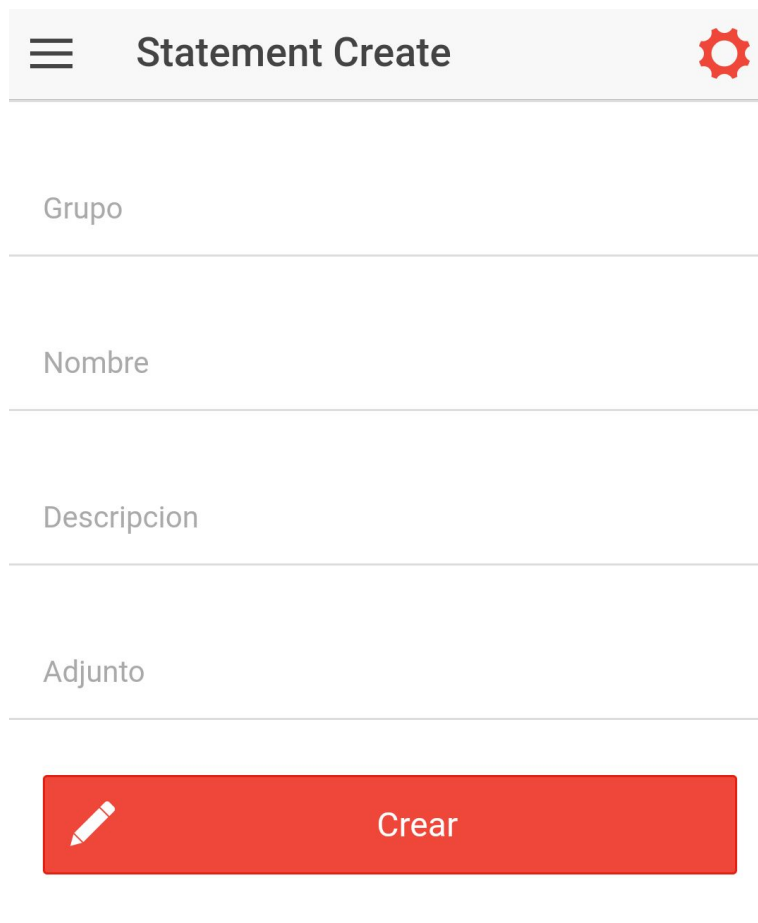
 **Buscar**

Figura 31: Buscar statements.

- **Grupo:** grupo o curso sobre cual buscar el statement. En nuestro ejemplo, el usuario de la cuenta “Moodle Info UNLP” puede estar registrado como alumno de la materia “Laboratorio de software” y en la materia “Matemática 4”. Si queremos filtrar los statements para recuperar sólo los relacionados a alguna de estas materias en particular tenemos que especificarlo en el filtro “**Grupo**”. El campo se autocompleta al comenzar a escribir sugiriendo los grupos que coincidan con la palabra escrita. Si no se conocen los grupos asociados a la cuenta, los mismos se pueden consultar desde la opción “Ver Grupos” del menú.
- **Fecha desde/hasta:** filtro para especificar la fecha de creación de los statements buscados. Ambas fechas son opcionales.
- **Etiquetas:** este filtro permite filtrar los statements etiquetados con las palabras indicadas en el mismo. Por ejemplo, podemos buscar statements etiquetados como “java programación”.

6.1.7 Crear Statement

El procedimiento para crear un statement, ya sea como respuesta (sub-statement) a otro statement o desde la opción “Crear Statement” es el mismo. Este puede apreciarse en la figura 32, la cual nos muestra un formulario para la carga del statement.



The image shows a web interface for creating a statement. At the top, there is a header bar with a hamburger menu icon on the left, the text "Statement Create" in the center, and a gear icon on the right. Below the header, there are four input fields, each with a horizontal line underneath: "Grupo", "Nombre", "Descripcion", and "Adjunto". At the bottom of the form is a prominent red button with a white pencil icon on the left and the text "Crear" in the center.

Figura 32 Crear statement.

Capítulo 7

7.1 Trabajos Futuros

Dado que este trabajo es solo con fines académicos, se han implementado las funcionalidades básicas que nos permitieron demostrar la utilidad del estándar estudiado en los capítulos de este documento, dejando para futuras implementaciones los siguientes aspectos.

7.1.1 Completar la funcionalidad provista por el estándar Tin Can

El desarrollo presentado es un prototipo el cual nos sirvió para demostrar no solo la utilidad del estándar Tin Can, sino a su vez la factibilidad de implementarlo en los sistemas de la Facultad de Informática, debido a esto solo se implementó la funcionalidad que fue requerida para dicho propósito, dejando para futuras implementaciones el resto de la funcionalidad.

7.1.2 Implementar single sign on

Dada a la multiplicidad de plataformas mediante las cuales el LRS puede interactuar, se debe pensar e implementar un sistema de single sign on para facilitar a los usuarios el accesos a los diversos sistemas.

El mismo permitirá loguearse a diversos sistemas de aprendizajes mediante el LRS. Este tipo de desarrollo necesitará el trabajo conjunto tanto de los administradores de LRS como así también de los administradores de los LMS.

7.1.3 Implementación de estadísticas

Un característica que aún no se ha explotado en el producto desarrollado es la capacidad de relevar datos estadísticos por parte de los administradores. Dado que todos los statements son registrados por el LRS, resulta sencillo implementar un módulo de estadísticas, para poder medir cuáles son los recursos más utilizados, los usuarios más activos, los recursos externos más compartidos, etc.

Dichos datos estadísticos serán de suma relevancia a los educadores para planificar los contenidos de las asignaturas y para medir el nivel de participación de los alumnos.

7.1.4 Mejoramiento de la interfaz de usuario del LRSapp

En el desarrollo del LRSapp no se contó con la asistencia de un diseñador gráfico con experiencia en interfaces de usuarios, recurrimos a una serie de templates que provee ionics. Se deja para futuros desarrollos la tarea de mejorar la interfaz de usuario de la aplicación móvil.

7.1.5 Pruebas del campo

El desarrollo solo se ha probado con usuarios ficticios, creados con el fin de testear la aplicación. Todas estas pruebas han sido orientadas a garantizar el correcto funcionamiento de la aplicación siguiendo el estándar TinCan. Se deja para futuras etapas del desarrollo la prueba de campo de la aplicación, lo cual implica que un grupo de alumnos de la facultad descargue la aplicación a sus celulares y la utilicen durante un determinado tiempo para registrar sus experiencias educativas. Una vez concluido el lapso de prueba los alumnos deberán hacer la devolución sobre los aspectos favorables y negativos de la aplicación.

7.2 Conclusión

A lo largo de los capítulos hemos visto, cómo la tecnología se ha ido introduciendo en el ámbito de la educación, hasta formar el concepto que hoy conocemos como e-learning. Este concepto que aporta a los alumnos una nueva manera de acceder a los conocimientos mediante el uso de la tecnología ha ido evolucionando a medida que ha evolucionado la tecnología. Esta evolución se ha regulado por medio de una serie de estándares, los cuales permiten modelar los objetos de aprendizajes y interrelacionar LMS's entre sí.

Como ha quedado en evidencia, el e-learning es un área en constante evolución, motivado principalmente por el surgimiento de las nuevas tecnologías, como así también por el uso y costumbre de los usuarios de dichas tecnologías.

Hemos hecho foco en el estándar Tin Can el cual es una evolución del estándar Scorm. Scorm ha sido utilizado ampliamente en los LMS's, tanto para modelar, como para compartir objetos de aprendizajes. Las ventajas significativas de Tin Can frente a su antecesor u otros estándares de aprendizajes es la adaptabilidad a nuevas tecnologías, como lo son teléfonos móviles y tablets, como también la posibilidad de agregar contenidos educativos que residen fuera de nuestros LMS.

Esta nueva forma de ver los objetos de aprendizajes, la cual nos permite ver recursos propios como ajenos, ha dado lugar a la creación de una nueva herramienta, el LRS, el cual permite trackear contenidos fuera de nuestros LMS's.

Las entidades educativas que adopten TinCan, estarán mejor adaptadas para los requerimientos que los alumnos actualmente exigen a una plataforma de e-learning, como es el acceso desde dispositivos móviles. A su vez dichos establecimientos están preparados para nuevas y desconocidas forma de e-learning venideras.

Referencias

- [1] García-Peñalvo, Una revisión actualizada del concepto de eLearning.
<http://repositorio.grial.eu/bitstream/grial/390/1/1138-9737-2015-0016-0001-0119-0144.pdf>
- [2] Introducción al e-learning
http://antia.fis.usal.es/sharedir/TOL/introlearning/21_definicion_de_elearning.html
- [3] Learning management system
<http://www.uv.es/ticape/pdf/CUE-LMS%20White%20Paper.pdf>
- [4] Learning Management System Feasibility Study
http://oscmoodlereport.files.wordpress.com/2010/08/osc_feasibility_study_full_report.pdf
- [5] Campus Virtual: la educación más allá del LMS
<http://www.uoc.edu/rusc/4/1/dt/esp/ortiz.pdf>
- [6] Uso de contenidos educativos digitales a través de sistema de gestión de aprendizaje (LMS) y su repercusión en el acto didáctico comunicativo.
<http://eprints.ucm.es/11631/1/T32372.pdf>
- [7] Moodle
<https://moodle.org/>
- [8] Meran
<http://www.cespi.unlp.edu.ar/meran>
- [9] Scorm
<http://scorm.com/tincanoverview/what-is-an-lrs-learning-record-store/>
- [10] Introducción al modelo de referencia SCORM
http://unizar.es/innovacion/coleccion1/archivos/PDF/49_SCORM_CURSO.pdf
- [11] TinCan
http://www.adlnet.gov/wp-content/uploads/2013/05/20130521_xAPI_v1.0.0-FINAL-correx.pdf
- [12] Learning Experiences Built Using the TLA Experience API
http://www.adlnet.gov/wp-content/uploads/2012/08/Hruska_ExAPI_LearningTools_iFest2012.pdf
- [13] The Learning Record Store (LRS)
<http://tincanapi.com/learning-record-store/>
- [14] Learning record store
<http://tincanapi.com/building-a-learning-record-store/>
- [15] Carlos Emilio Biscay, Los estándares de e-learning,
<http://www.palermo.edu/ingenieria/downloads/CyT5/CYT506.pdf>
- [16] Learning Object Metadata

https://en.wikipedia.org/wiki/Learning_object_metadata

[17] Harman, K. (2007). *Learning objects: standards, metadata, repositories, and LCMS*. Informing Science.

[18] Uso de estándares aplicados a TIC en Educación.

<http://ares.cnice.mec.es/informes/16/contenido/39.htm>

[19] Aviation Industry Computer-Based Training Committee

<https://www.trainingindustry.com/taxonomy/a/aicc.aspx>

[20] Advanced Distributed Learning

<http://www.adlnet.org/>

[21] IMS Global Learning Consortium

<https://www.imsglobal.org/>

[22] The Learning Technology Observatory

<http://www.cen-ltso.net/>.

[23] IEEE Learning Technology Standards Committee (LTSC)

<https://ieee-sa.centraldesktop.com/ltsc/>

[24] Institute of Electrical and Electronics Engineers

<http://www.ieee.org/index.html>

[25] A. Wiley, "LEARNING OBJECT DESIGN AND SEQUENCING THEORY", Brigham Young University, 2000.

<https://opencontent.org/docs/dissertation.pdf>

[26] R. Polsani, "Use and Abuse of Reusable Learning Objects", Learning Technology Center, University of Arizona, USA.

<https://journals.tdl.org/jodi/index.php/jodi/article/view/89/88>

[27] Szabo, Micheal; Flesher, K. (2002). "CMI Theory and Practice: Historical Roots of Learning Management Systems". *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2002 (White Paper)* (Montreal, Canada: In M. Driscoll & T. Reeves (Eds.)): pp. 929–936. ISBN 1-880094-46-0.

[28] Ellis, Ryann K. (2009), *Field Guide to Learning Management Systems*, ASTD Learning Circuits

[29] El estándar SCORM para EaD

<http://www.mrebollo.es/pubs/tesina.pdf>

[30] ADL Scorm

<https://www.adlnet.gov/adl-research/scorm/>

[31] SCORM, un aliado para el e-learning.

<http://cursosyformaciononline.es/destacados/scorm-elearning>

- [32] Tin Can Api: Connecting Learning Experiences
<http://tincanapi.com/wp-content/assets/Onion/Layers-of-the-Tin-Can-Onion.pdf>
- [33] Tin Can Api
<http://tincanapi.com/>
- [34] XApi
<https://elearningindustry.com/?s=tin+can+api>
- [35] Tin Can API, porque aprendemos en cualquier lugar
<https://puntomov.com/2013/06/18/que-es-tin-can-api/>
- [36] TIN CAN: Registro de experiencias de aprendizaje
<http://www.webdoor.com.ar/blog/tin-can-registro-de-experiencias-de-aprendizaje/>
- [37] Tin Can Capabilities
<http://scorm.com/tincancapabilities/>
- [38] Especificación Tin Can, parte 1
<https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-About.md#partone>
- [39] Especificación Tin Can, parte 2
<https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md#parttwo>
- [40] Especificación Tin Can, parte 3
<https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Communication.md#partthree>
- [41] Leonard Richardson, "RESTful Web APIs", ISBN-13: 978-1449358068
[http://sd.blackball.lv/library/RESTful_Web_APIs_\(2013\).pdf](http://sd.blackball.lv/library/RESTful_Web_APIs_(2013).pdf)
- [42] Building a Learning Record Store
<https://tincanapi.com/building-a-learning-record-store/>
- [43] The Enterprise Learning Ecosystem
<https://tincanapi.com/ecosystem/>
- [44] Spring
<https://spring.io/>
- [45] MySQL
<https://www.mysql.com/>
- [46] Tomcat
<http://tomcat.apache.org/>
- [47] Marko Gargenta, "Learning Android ", 2011. ISBN: 978-1-449-39050-1
<http://gsl.mit.edu/media/programs/south-africa-summer-2015/materials/learningandroid.pdf>
- [48] Google Play Store
<https://play.google.com/store>

[49] M. Panhale, "Beginning Hybrid Mobile Application Development". ISBN 978-1-4842-1315-5

[50] Apache Cordova

<https://cordova.apache.org/>

[51] A.Ravulavaru, "Learning Ionic", 2015, ISBN 978-1-78355-260-3.

<http://pepa.holla.cz/wp-content/uploads/2016/12/Learning-Ionic.pdf>

[52] Ionic Framework

<https://ionicframework.com/>

[53] AngularJS Framework

<https://angularjs.org/>

[54] Manual de introducción a Ionic.

<https://ajgalleo.gitbooks.io/ionic/content/>

[55] Packaged Web Apps (Widgets)

<https://www.w3.org/TR/widgets/>