

The logo of the University of La Laguna (ULL) consists of the letters 'ULL' in a stylized, purple, sans-serif font. The 'U' is a single continuous stroke, while the 'L's are also connected. A thin horizontal line is positioned below the letters.

Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Gamificación del proceso de enseñanza-aprendizaje sobre un LMS integrando xAPI

*Teaching-learning process gamification on
a LMS using xAPI*

Daniel Afonso González

Escuela Superior de Ingeniería y Tecnología

Trabajo fin de grado

Dña. **Carina Soledad González González**, con N.I.F. 54.064.251-Z profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora.

D. **Alberto Mora Carreño**, con N.I.F. 78.715.985-A doctorando adscrito al Departamento de Computer, Multimedia and Telecommunication Studies de la Universitat Oberta de Catalunya como co-tutor.

C E R T I F I C A N

Que la presente memoria titulada:

Gamificación del proceso de enseñanza-aprendizaje sobre un LMS integrando xAPI

ha sido realizada bajo su dirección por D. **Daniel Afonso Gonzalez**, con N.I.F. 78.609.122-K.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de Septiembre de 2015.

Agradecimientos

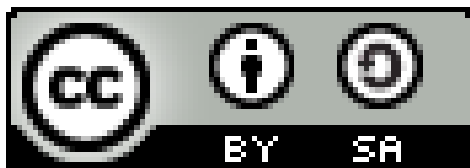
A mi familia por su apoyo incansable. En especial a mi madre que me dio la vida y me alienta en cada paso que doy.

A mis tutores por su interés.

A mis profesores y compañeros del ciclo formativo de Desarrollo en Aplicaciones Informáticas, por devolverme la ilusión de programar y animarme a terminar mis estudios.

A todos mis compañeros del grado de ingeniería informática, con los que tan buenos momentos he pasado.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional.

Resumen

Este proyecto nace con la intención de automatizar la recogida de información relevante para la gamificación del proceso de enseñanza-aprendizaje existente en sistemas formales de aprendizaje, mediante el uso de la Experience API, a partir de ahora xAPI. Se espera que gracias a esta tecnología, se pueda automatizar la gamificación de entornos formales e informales, de manera sencilla y transparente al usuario.

Gamificar un entorno, consiste en convertir tareas monótonas o aburridas en otras más atractivas para el usuario. Haciendo uso de esta tecnología podemos recopilar muchas de las conductas y/o datos de los usuarios de estas plataformas. Para esto se ha hecho uso de un servidor LRS (Learning Record Store) que permite registrar las actividades realizadas por un usuario en un entorno educativo, en nuestro caso, nuestro entorno es Open edX. Si unimos además un motor de gamificación, permite poder realizar una gamificación de algunos de los procesos de ese entorno educativo. Gracias a esto, se ha conseguido una pequeña demostración de un entorno Open edX gamificado.

Palabras clave: gamificación, enseñanza-aprendizaje, xAPI, LRS, LMS, Open EDX.

Abstract

This project starts with the automation of the collection of information relevant to the gamification of the teaching-learning processes that exists in formal learning systems, using the Experience API, now known as xAPI. It is expected that thanks to this technology, the gamification of formal and informal environments can be automated in an easy and user transparent way.

The gamification of an environment, consist in converting monotone and boring tasks into others more attractive to the user. Using this technology, we can gather many of the behaviours and/or data of the users of these platforms. To this end, an LRS (Learning Record Store) server has been used, which allows the recording of the activities carried out in an educational environment, which in our case is Open edX. If, on top of that, a gamification engine is added, it can gamify some of the processes of this educational environment. Thanks to this, a little demonstration of a gamified Open edX environment has been completed.

Keywords: gamification, teaching-learning process, xAPI, LRS, LMS, Open EDX.

Índice general

Capítulo 1. Introducción	9
Capítulo 2. Estado del Arte	11
2.1 Gamificación	11
2.2 Sistemas de Gestión del Aprendizaje	14
2.3 Tipos de Sistemas Gestores del Aprendizaje	15
2.4 Importancia de los Sistemas Gestores del Aprendizaje	16
2.5 Open edX	17
2.6 Experience API	19
2.7 Motores de gamificación	21
Capítulo 3. Objetivos del Proyecto	22
3.1 Identificación de los procesos a gamificar.	24
3.2. Implementación del servidor LRS.	25
3.3 Integración de Experience API con Open edX.	27
3.4 Integración de un motor de gamificación con Captain Up.	29
Capítulo 4. Conclusiones y líneas futuras.	30
Capítulo 5. Summary and conclusions.	32
Capítulo 6. Presupuesto.	34
Bibliografía	36
Capítulo 7. Apéndices	37

Índice de Figuras

Figura 1. Arquitectura de Open edX.	18
Figura 2. Arquitectura de xAPI.	19
Figura 3. Arquitectura del proyecto.	22
Figura 4. Interfaz gráfica servidor LRS	25
Figura 5. Portada de EDX tras el login de usuario	28
Figura 6. Vista del desplegable de Captain Up	29

Índice de Tablas

Tabla 1. Clasificación de tipos de sistemas gestores del conocimiento.	15
Tabla 2. Niveles definidos para entorno gamificado.	24
Tabla 3. Insignias definidas para entorno gamificado.	24
Tabla 4. Tabla resumen del presupuesto general.	35

Capítulo 1. Introducción

La aplicación de las tecnologías de la información, en el ámbito académico y la expansión del acceso a internet ha supuesto una auténtica revolución, permitiendo no solo una mayor interacción entre los distintos agentes del sistema educativo, sino también posibilitando a las diferentes universidades el poder ampliar su oferta docente través de plataformas de aulas virtuales. Estos sistemas de aulas virtuales se hicieron cada vez más complejos y, para simplificar el tratamiento de los datos, comenzaron a surgir los llamado sistemas gestores del conocimiento o Learning Management Systems (LMS de ahora en adelante, por sus siglas en inglés). Estos son capaces, de entre otras funciones, de entregar el contenido del curso a los usuarios, registrar y administrar a los usuarios, gestionar los distintos casos prácticos del curso, y un largo etcétera. Sin embargo, estos sistemas gestores usaban formatos propietarios que impedían el poder compartir estos contenidos entre distintos LMS's, por ello surge SCORM (Sharable Content Object Reference Model)^[2], que establece una serie de estándares para la creación de documentos pedagógicos estructurados.

A partir del nacimiento de SCORM se ha desarrollado en los últimos años la TinCan API, rebautizada en sus últimas versiones como Experience API. Esta API nos permite no solo controlar el comportamiento de los usuarios dentro un LMS tradicional, sino también en otros ámbitos más relacionados con el ocio, por lo que es muy interesante su aplicación en gamificación. Este último concepto en palabras de Sebastian Deterding, *“es el uso de mecánicas y diseño de videojuegos en entornos no lúdicos”*^[1].

Esta gamificación podemos usarla tanto para la calificación del trabajo del alumno, como para aumentar la motivación del mismo al realizar tareas que le premian de manera inmediata con el uso de insignias y puntuaciones.

Un ejemplo de proceso gamificado podría ser un ejercicio que consistiera en resolver una batería de ecuaciones en un tiempo determinado, puntuándolas en función del tiempo transcurrido en su resolución y su grado de dificultad. Otro ejemplo sería el publicar una clasificación de usuarios en función de los libros que han leído en los últimos meses.

Ambas acciones están encaminadas a fomentar un comportamiento definido previamente por el docente para la consecución de objetivos en su alumnado.

Capítulo 2. Estado del Arte

2.1 Gamificación

La gamificación se ha convertido a día de hoy en una poderosa herramienta para fidelizar clientela en el mundo del marketing, una definición formal de la misma según Sebastian Deterding es *“el empleo de mecánicas de juego en entornos y aplicaciones no lúdicas con el fin de potenciar la motivación, la concentración, el esfuerzo, la fidelización y otros valores positivos comunes a todos los juegos.”* ^[1]

Esta idea no es del todo novedosa ya que lleva años siendo usada en otros campos, un ejemplo claro y sencillo sería el sistema de pegatinas y premios que suelen utilizar los profesores de educación infantil con sus alumnos, premiando el esfuerzo y reforzando las buenas conductas. Si esto lo trasladamos al ámbito universitario podemos otorgarles puntos y premios al alumnado por leer un documento, ver un vídeo instructivo, escribir un comentario o participar en una actividad; motivándolos en cualquier caso a la consecución de objetivos que el profesor se ha propuesto.

Aunque no se trate de algo totalmente nuevo, sí que el auge de los videojuegos y la eclosión de comunidades online han supuesto un aumento del uso de esta técnica y abre la puerta a nuevos modos de aprendizaje.

Para comprender mejor la gamificación tenemos que entender qué gamificar un entorno o un proceso no es lo mismo que diseñar un juego, a pesar de que muchos de los conceptos utilizados en gamificación vengan directamente del diseño de juegos de carácter general. Según Robin Hunicke ^[3], cuando gamificamos un entorno o una tarea debemos tener en cuenta tres conceptos:

- Mecánicas de juego.
- Dinámicas de juego.
- Estéticas de juego.

Las mecánicas de juego son el componente más pequeño que encontramos en cualquier juego, y a su vez las más importantes, ya que son ellas las que determinarán los distintos elementos del juego a nivel de datos y algoritmos.

Las dinámicas de juego se encargan de definir el comportamiento de las distintas mecánicas de juego al interactuar los usuarios con estas.

Las estéticas de juego, por último, son aquellas en las que definimos las distintas respuestas emocionales que esperamos despertar en un usuario cuando interactúa con el sistema gamificado.

Numerosos estudios^{[4][5]} han demostrado que gracias al uso de la gamificación, los resultados académicos de los alumnos se ven incrementados. Ya que al entrar en juego el factor diversión, la motivación se dispara, a la vez que se reduce el estrés y el miedo al error. Además durante la gamificación el usuario obtiene un *feedback*, que le permite aprender mediante prueba y/o error. Esto favorece la asimilación de contenidos por parte del alumno y no solo su memorización como ocurre en muchos casos.

Por esta razón se han empezado a gamificar los procesos educativos, algunos con mejor o peor éxito, pero todos con un gran talón de Aquiles, que es la captura de lo que hace el usuario fuera de un entorno formal LMS. Esta carencia, que se intenta paliar con estándares como el de TinCan API, que ya hemos dicho que es una evolución del estándar SCORM, nos permite por primera vez salir del entorno educativo y poder capturar así cualquier acción que realice el usuario fuera de las plataformas educativas tradicionales. De esta manera, aumentamos de manera notable, la cantidad de información obtenida, utilizándola para mejorar el proceso de gamificación de cualquier aprendizaje, tanto formal como informal.

La potencia de la gamificación se hace factible en los casos de éxito de empresas que ya lo han utilizado. Un ejemplo es el del Club Psych de NBC/Universal, que vio cómo el número de páginas vistas aumentaba un 130% y los visitantes recurrentes en un 40% tras implementar una campaña de Gamificación en el año 2011. A través de una serie de incentivos, aumentó su retención, incrementando sus ingresos, que llegaron a dispararse y situarse en casi 3 millones de usuarios registrados.

Otras empresas utilizan la gamificación para motivar a sus trabajadores mediante un sistema de puntuaciones, logros y retos. Un ejemplo claro de esto lo constituye la consultora Deloitte ^[6], que implementó mecánicas de gamificación con clasificaciones e insignias en su academia para ejecutivos. Con ello consiguieron un incremento del 37% de usuarios habituales en su portal de formación, así como un aumento del 40% en visitantes semanales.

2.2 Sistemas de Gestión del Aprendizaje

Cuando hablamos de Sistemas de Gestión del Aprendizaje (SGA), nos referimos a un sistema web capaz de administrar, distribuir y controlar las actividades de formación virtuales de una organización.

Estos sistemas son los encargados de administrar la gestión de usuarios, el acceso a los recursos de formación y las actividades de formación. Normalmente estos sistemas no se encargan de generar contenido propio, sino que se caracterizan por gestionar contenidos creados por diversas fuentes, aunque algunos sistemas incluyen un “Sistema de Gestión del Contenido del Aprendizaje” o LCMS (Learning Content Management System) de sus siglas en inglés. Encontramos ejemplos de este tipo en las herramientas de creación de cursos de Moodle o el edXStudio. Todos estos sistemas siguen las normas de los distintos estándares, los cuales garantizan lo siguiente^[11]:

- Interoperabilidad
 - Capacidad de varios SGA para trabajar entre sí.
- Reusabilidad
 - Todo el contenido generado dentro de un SGA debería poder ser reusado en cualquier otro Sistema Gestor del Aprendizaje.
- Gestion
 - Todo Sistema Gestor del contenido debe permitir acceder a la información correcta tanto de los usuarios como del contenido.
- Accesibilidad
 - El Sistema Gestor del Contenido debe permitir un fácil acceso a los contenidos, en unos tiempos concretos.
- Durabilidad
 - El Sistema Gestor debe poder escalar con el tiempo, a la vez que superar la obsolescencia, mediante actualizaciones constantes que incorporen nuevas tecnologías.

2.3 Tipos de Sistemas Gestores del Aprendizaje

Los sistemas gestores del aprendizaje se pueden clasificar en dos grandes bloques:

- OpenSource
- Privativos

Open Source	Software privativo
Moodle	Atutor
Open edX	Blackboard
Claroline	WebCT

Tabla 1. Clasificación de tipos de sistemas gestores del conocimiento.

Las ventajas e inconvenientes de unos sobre otros vienen a ser las ya conocidas del software libre sobre el privativo, mientras que en uno los costes de implantación son bajos y tenemos acceso al código fuente de los proyectos, en el otro, tenemos una mayor adaptación de los sistemas privativos a las necesidades del usuario, siempre y cuando este sea capaz de afrontar el coste de su implantación.

En los últimos años ha habido un auge de las aplicaciones de código abierto, como Moodle, que está aumentando cada vez más su nicho de mercado, según consta en las analíticas de la web capterra ^[10]. El aumento del éxito de aplicaciones código abierto hicieron que en Abril de 2013, surgiera OpenEdx, fruto de la colaboración de la Universidad de Harvard y el Instituto Tecnológico de Massachusetts, proyecto al que se sumaría tiempo después Google y otras Universidades, y del que hablaremos en siguientes capítulos.

2.4 Importancia de los Sistemas Gestores del Aprendizaje

En los últimos años el incremento de Cursos en Línea Masivos y Abiertos o Mooc's, según sus siglas en inglés (Massive Open Online Course), ha potenciado el uso de los sistemas gestores del aprendizaje de código abierto. Son múltiples las plataformas online que ofertan estos cursos, siendo las más conocidas edX, Coursera y MiriadaX. La acogida ha sido tal que Coursera ya cuenta con más de once millones de usuarios lo que la convierte en la mayor universidad del mundo. Ofertando más de mil cursos provenientes de más de 100 universidades de todo el mundo. edX, en menos de 3 años de vida, ya cuenta con más de 4 millones de usuarios, y MiriadaX, con más de millón y medio de usuarios en todo el mundo^[5].

Aunque pudiera parecer que estos datos han coronado a los MOOCS como los reyes de la educación online, existen datos desalentadores, ya que diversos estudios demuestran que la tasa de abandonos se encuentra entre el 75% y el 95% de los matriculados^{[7][9]}. Muchas veces por falta de preparación del alumnado que se inscribe y otras por falta de motivación. Es por esto último, que en este proyecto, nos centraremos en la gamificación de Open edX. Ya que como se ha constatado en capítulos anteriores, un uso correcto de las herramientas de gamificación pueden aumentar la fidelización y el uso de una plataforma de manera espectacular, sirviendo de impulso en la lucha contra la enorme cifra de abandonos en los MOOCS.

2.5 Open edX

Open edX es una plataforma de cursos en línea masivos y abiertos (MOOC), desarrollada en Django. Se trata de un proyecto opensource impulsado por el Instituto Tecnológico de Massachusetts, la Universidad de Harvard y la Universidad de Stanford, a la que se han ido sumando a lo largo de los años otras instituciones de diversos países como la Escuela Politécnica Federal de Lausana en Suiza o la Universidad Técnica de Delft.

La plataforma se divide en varios elementos bien diferenciados, siendo los más importantes su LMS y CMS (*Content Management System*), que son similares a lo que encontramos en otros sistemas como MOODLE. Por un lado tenemos un LMS o sistema gestor del conocimiento, que es el encargado de mostrar a los alumnos las diferentes tareas y cuestionarios, es decir, define el entorno de trabajo de todos los alumnos. Por otro lado, un CMS o sistema gestor de cursos, donde el profesorado diseña los cursos para la plataforma, permitiendo elegir entre una gran variedad de tareas. Todos estos datos, a su vez, son almacenados en una base de datos MongoDB, mientras que la información de los usuarios se guarda en una base de datos MySQL.

Además de estos sistemas tenemos otros auxiliares, como podemos ver en la figura 1. Disponemos de foros que se encuentran desarrollados en Ruby, un sistema de certificados también desarrollado en Django, o las distintas API basadas en REST, que nos permiten obtener información de los usuarios o cursos. En definitiva, Open edX se trata de un sistema abierto que utiliza múltiples tecnologías, aunque las más importantes sean Python y JavaScript.

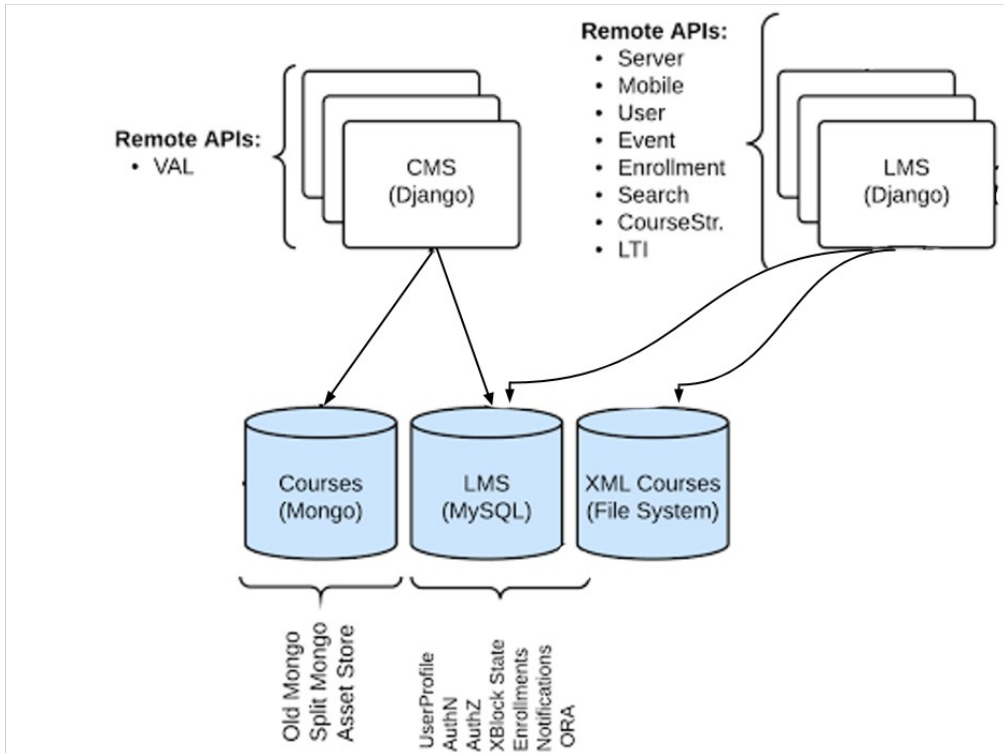


Figura 1: Arquitectura de OpenEDX

2.6 Experience API

La Experience API o xAPI es un estándar de reciente creación para las aplicaciones de e-learning, el cual nos permite almacenar de manera estructurada las interacciones de un usuario ante distintas actividades de aprendizaje, ya sean estas en entornos tradicionales de la enseñanza online como los LMS's o nuevas aplicaciones fuera de estos entornos de e-learning.

Como podemos ver en la figura 2, existen dos actores principales en Experience API, por un lado tenemos a los proveedores de actividades, que son los encargados de registrar la experiencia del usuario. Estos pueden ser cualquier dispositivo o software que nos permita aprender con independencia de dónde nos encontremos. Y por otro lado tenemos los LRS (Learning Record Store) que son los encargados de almacenar las distintas actividades proveídas por los proveedores de actividades. Estos servidores LRS son independientes unos de otros, aunque pueden compartir información entre ellos de los distintos usuarios.

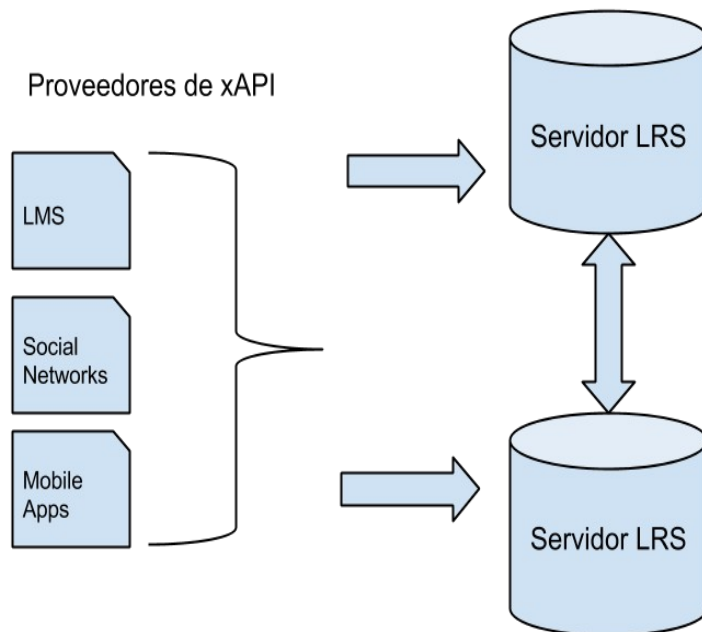


Figura 2. Arquitectura de xAPI

La principal filosofía en la que está basada xAPI es que el aprendizaje no se da solo dentro de los LMS's sino que se pueda dar en cualquier plataforma y/o dispositivo esté o no conectado a la red. En este segundo caso las sentencias de xAPI deberán almacenarse en el propio dispositivo hasta que este pueda enviarlas a un servidor LRS.

2.7 Motores de Gamificación

Un motor de gamificación es una solución informática que permite definir una serie de recompensas a los usuarios de una determinada plataforma. Esto se hace con el uso de insignias, puntos o misiones, que nos permiten acceder a dichas recompensas. A la hora de diseñar un sistema gamificado se deben tener en cuenta las limitaciones de los distintos motores de gamificación, como podrían ser tipo de recompensas que ofrecer a los usuarios o las diferentes mecánicas permitidas por los mismos. Por lo tanto debe hacerse un análisis exhaustivo de los distintos motores de gamificación que existen en el mercado, para así poder valorar entre las distintas opciones.

Para este proyecto, en concreto, se decidió apostar por Captain Up, al permitirnos, con su versión gratuita, acercarnos a los requisitos que necesitábamos para nuestro sistema gamificado. Anteriormente se planteó la posibilidad de utilizar otros motores como Mambo.io o Gamify, pero pronto se descartaron al ser imposible acceder a dichos motores en su acceso anticipado.

Captain Up, en su versión gratuita, nos permite definir hasta ocho niveles y 10 insignias, además de utilizar su API para HTML5 y Javascript, así como poder realizar un seguimiento de las acciones de los usuarios.

Capítulo 3. Objetivos del Proyecto

Teniendo en cuenta lo presentado con anterioridad, se consideró la siguiente serie de objetivos para este proyecto:

- Identificación de los procesos a gamificar.
- Implementación de un LRS (Learn Record Store).
- Integración de Experience API con OpenEDX.
- Integración de un motor de gamificación.

Quedando reflejado en una estructura similar a la que se muestra en la Figura 3.

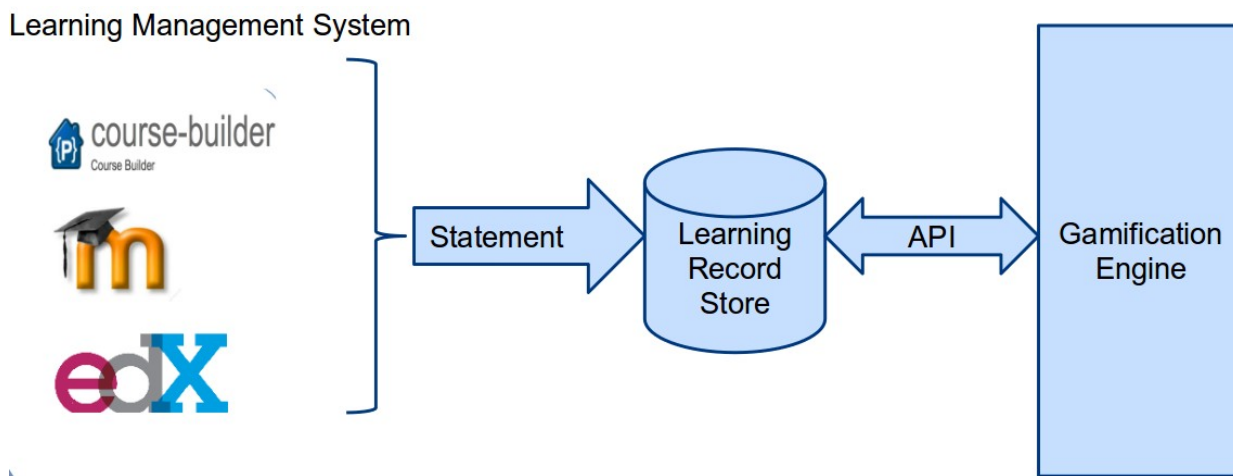


Figura 3. Arquitectura del Proyecto

Por un lado tenemos los entornos LMS que se encargan de suministrar al servidor LRS las sentencias sobre las distintas acciones que realizan los usuarios dentro de estos entornos. Estas sentencias son analizadas por el servidor LRS, que comprueba que son correctas y aquellas que han sido definidas por el programador son remitidas al motor de gamificación para que sean tenidas en cuenta a la hora de gamificar el sistema. Se tomó la decisión de adoptar esta arquitectura, con el fin de garantizar que las tres partes que la componen, fueran independientes las unas de las otras. Permitiendo así poder añadir nuevos módulos a la arquitectura sin que esta sufra cambios drásticos, por ejemplo podríamos incluir un nuevo LMS

o utilizar varios servidores LRS, o cambiar de motor de gamificación, sin que los protocolos de comunicación de las demás partes cambien .

3.1 Identificación de los procesos a gamificar

A la hora de gamificar un sistema debemos identificar qué tareas deseamos gamificar y cómo vamos a hacerlo. Como ya hemos visto en el apartado dedicado a gamificación es muy importante definir las dinámicas de juego, en este trabajo fin de grado, al ser una aproximación a la gamificación, hemos definido una serie de mecánicas basadas en insignias y niveles. Estos se consiguen según se van realizando distintas actividades dentro de la plataforma educativa, como por ejemplo visualizar los vídeos de las distintas lecciones o la realización de cuestionarios. A continuación podemos ver la relación de niveles y puntos así como las de acciones e insignias.

Nivel	Puntos necesarios
Novato	0
Aprendiz	25
Veterano	200
Maestro	500
Leyenda	1000

Tabla 2. Niveles definidos para entorno gamificado.

Insignia	Descripción
Vigilante	Visualiza 10 videos
Sabio	Lee 10 libros
Instruido	Contesta 10 cuestionarios
Experto	Responde a 10 preguntas de los usuarios

Tabla 3. Insignias definidas para entorno gamificado.

3.2 Implementación del servidor LRS

Aunque inicialmente se optó por desarrollar todo un servidor LRS en Django desde cero, pronto hubo que desecharla al disponer ya de una versión libre, también escrita en Django, por parte de la ADL (*Advanced Distributed Learning*) institución impulsora de la xAPI. Gracias a la cual, una vez resueltos algunos problemas de librerías y migración de Django 1.4 a su última versión 1.8, permitió disponer de un servidor LRS operativo en poco tiempo, ahorrando muchas horas de desarrollo.

Para poder enviar oraciones a este servidor, primero debemos crear un usuario, para así poder identificar correctamente las distintas fuentes de datos que llegarán a nuestro servidor. Para hacer esto basta con tener el servidor operativo y acceder a él a través de la dirección que hayamos especificado a nuestro servidor de Django, y nos aparecerá una interfaz web como la siguiente:

ADL LRS

Username Password Sign in

Task me on GitHub

Welcome to the ADL Learning Record Store

EXPERIENCE API

This Learning Record Store (LRS) is a reference implementation of an LRS as described in the [Experience API Specification](#).

Current LRS Stats

Users	Statements	Agents	Verbs	Activities
2	167	12	8	20

Sign up
Sign up for HTTP Basic Credentials
Create a username and supply an email address with it
[Register](#)

GitHub Examples
Visit our GitHub prototype page to find some of the updated 1.0.1 prototypes. These prototypes can be configured to report to an LRS. See the [Readme](#) for more information.
[GitHub Prototypes](#) [Launch](#)

Statement Validator
Want to make sure your statement is valid? Check it here.
[Validator](#)

OAuth Clients

- By default, only HTTP Auth is enabled. To enable OAuth, set the 'OAUTH_ENABLED' flag to true in settings.py
- Visit our client registration page
- Submit the name and description of the client you wish to authorize
- Once submitted, you will receive a client ID and client secret. Use these with the appropriate OAuth headers to perform the initial handshake to gain your temporary token. (/XAPI/OAuth/initiate) **NOTE: If you are not familiar with OAuth, please consult the spec and the OAuth 1.0 RFC**
- You'll then receive your request token. Use this when visiting the authentication page (/XAPI/OAuth/authorize). An LRS OAuth ID will be issued to you for you to POST back to the authentication page from your client
- Once the client is authenticated, send the appropriate OAuth headers and values from your token to gain an access token. (/XAPI/OAuth/token)

Figura 4. Interfaz gráfica servidor LRS

Como ya hemos mencionado anteriormente en el apartado sobre xAPI, este consta de dos partes fundamentales: un proveedor y un servidor LRS. Los proveedores de actividades pueden ser desde un dispositivo móvil a un LMS tradicional. Estos proveen a uno o varios servidores LRS con declaraciones que siguen la forma sujeto-verbo-objeto. (Véase apéndice A.1) Un ejemplo sencillo sería una frase como: *Yo hice esto*. Estas oraciones llegan a nuestro servidor LRS, que se encarga de almacenarlas en la base de datos del servidor. Posteriormente nos permitirá poder realizar búsquedas de las actividades que realiza un usuario, y procesarlas para comprobar que las peticiones están correctamente formadas.

3.3 Integración de Experience API con Open edX

Para gamificar las actividades que definimos en el punto 3.1, hemos hecho uso de un script de Python que se ejecuta en el servidor de Open edX. Dicho script se encarga de vigilar los cambios en los ficheros log de Open edX. Ante la detección de determinados cambios definidos en su código formatea una oración de xAPI, en la cual figura información del suministrador y del usuario que realiza dicha acción, como se puede ver en el apéndice A.1. Estas peticiones se envían a una cola en la que se almacenan, hasta que la misma se llene o pase un determinado tiempo, momento en el cual, son enviadas al servidor LRS que hayamos especificado. Esto se hace con el fin de optimizar el uso de la red. Este programa también pertenece a la ADL, aunque también se ha debido proceder a su modificación para poder obtener una integración óptima con el resto del sistema.

3.4 Integración de un motor de gamificación con Captain Up

Para integrar el motor de gamificación que escogimos en el punto 2.7 con la plataforma Open edX se han tenido que realizar una serie de modificaciones, todas ellas siguiendo las detalladas instrucciones que se encuentran en la documentación de Captain Up^[8]. Primero se han modificado las plantillas de Open edX para incluir un desplegable, gracias al cual el usuario podrá acceder a su usuario de Captain Up y consultar sus puntuaciones e insignias. El código añadido puede verse en el apéndice A.2.

Una vez realizada esta conexión entre Open edX y Captain Up, hubo que conectar nuestro servidor LRS a Captain Up, para así poder otorgar las distintas puntuaciones a los usuarios según las tareas que definimos con anterioridad. Cada vez que el servidor recibe la petición y comprueba que esta es correcta sintácticamente, haciendo uso de las señales de Django, se dispara el código que puede verse en el apéndice A.3, que da formato a los datos que enviamos al servidor de Captain Up. Cuando estos datos llegan al motor de gamification, este otorga los puntos definidos, y el usuario puede ver de manera inmediata su puntuación, como puede apreciarse en las figuras 5 y 6.

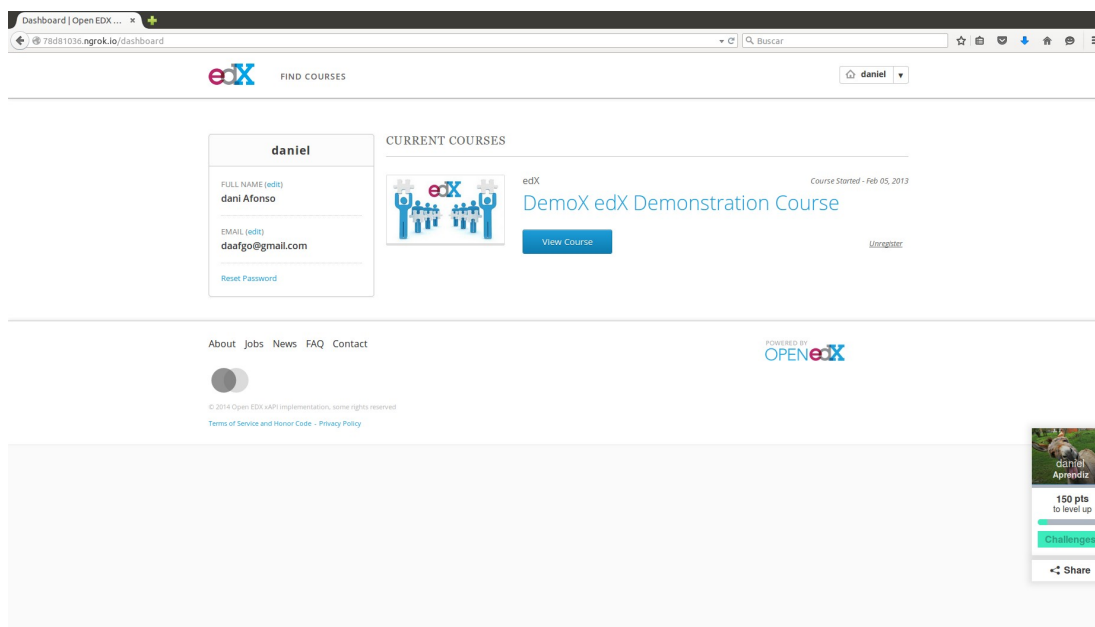


Figura 5. Portada de edX tras el login del usuario

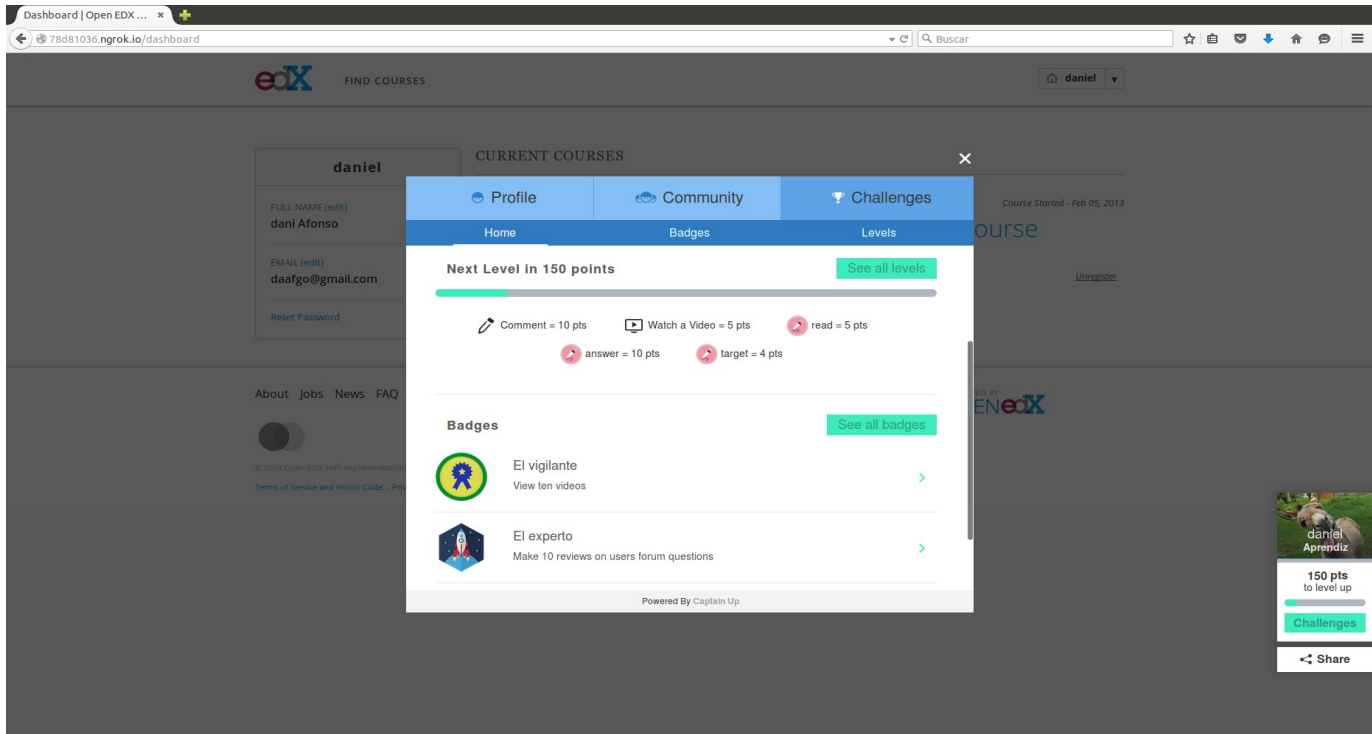


Figura 6. Vista del desplegable de Captain Up

Capítulo 4. Conclusiones y líneas futuras

El desarrollo de este proyecto se ha realizado en varias fases. La primera, fue la del análisis en la que se marcaron los objetivos del mismo: Identificar los procesos a gamificar, implementación de un LRS, integración de Experience API con Open edX y la integración de un motor de gamificación.

En una segunda fase se diseñó la arquitectura del proyecto y, en el transcurso de la misma se modificó el desarrollo del servidor LRS propio desarrollado en Django, puesto que se encontró una versión del mismo también desarrollada en Django por la ADL (*Advanced Distributed learning*), organización encargada del estándar xAPI.

En una tercera fase se usó un bridge desarrollado por la propia ADL para conectar Open edX y nuestro servidor LRS. Al realizar esta conexión, hubo que hacer alguna modificación con el fin de adecuar los parámetros necesarios para la gamificación del sistema. Una vez actualizadas algunas librerías y migrados el bridge y el LRS a la nueva versión de Django, se pudo filtrar qué acciones de usuario serían gamificadas y enviadas al motor de gamificación, haciendo uso del sistema de señales de Django. En cuanto al motor de gamificación usado, se ha optado por utilizar la versión gratuita de Captain Up, que pese a sus limitaciones, es suficiente para los objetivos de gamificación implementados en este proyecto.

Para el futuro se hace necesario desarrollar un motor de gamificación propio que permita el uso de mecánicas más complejas, o encontrar algún motor de terceros que no se encuentre tan limitado en cuanto a elementos como insignias o niveles. También para mejorar la gamificación y la experiencia del usuario se debería hacer uso de la gran cantidad de datos del usuario que se encuentran almacenados en nuestro servidor LRS, gracias a xAPI. Estos datos permitirán conocer mejor la conducta de nuestros usuarios, gracias a la cual, podríamos personalizar la información a mostrar en pantalla para

cada uno de ellos. Esto mismo ya se hace en algunos juegos de ordenador, así como el uso de un sistema recomendador.

Se debería plantear qué tipos de datos del usuario son los más relevantes a tener en cuenta para llevar a cabo estos sistemas de personalización y/o recomendación, así como desarrollar los componentes que nos permitan registrarlos. Para ello, no sólo debemos centrarnos en las distintas tecnologías web actuales, sino también en toda la gama de dispositivos móviles, que nos permiten aprender en cualquier momento con independencia de nuestra localización.

Capítulo 5. Summary and Conclusions

The development of this project was implemented in several phases. The first was the analysis in which the objective was marked: Identifying gamification processes, implementation of a LRS, integration with Open Experience API edX and integration of a gamification engine.

In the second phase the project architecture was designed and while it was being done, the development of an own LRS server in Django was scrapped, since a version of it was found, also developed in Django by ADL (Advanced Distributed Learning) the organization in charge of the xAPI standard.

In the third phase, a bridge developed by the ADL itself was used to connect the LRS server and Open edX. While implementing said connection, we had to make some modifications in order to adapt the required parameters for the gamification system. Once some libraries got updated, the bridge and LRS server got migrated to the new version Django, the user actions which could be gamified were filtered and sent to the gamification engine, using the Django signals system. As for the used gamification engine, the chosen one was the free version of Captain Up, which despite its limitations, is sufficient for the purposes of gamification implemented in this project.

For the future, it is necessary to develop proprietary gamification engine allowing the use of more complex mechanics, or finding a third party engine that is not as limited in term of items such as badges or levels. Also, to improve the gamification and the user experience, the big amount of data stored in our LRS server thanks to xAPI should be used. These data will allow to gain better understanding of the behavior of our users, which we could use to customize the information displayed on the screen for each user. The same is already done in some computer games, as well as the use of a recommender system.

It should be proposed what types of user data are more relevant to consider to carry out these personalization and / or recommendation systems, and develop components that allow us to register them. To do

this, we must not only focus on the various current web technologies, but also in the entire range of mobile devices that allow us to learn anytime regardless of our location.

Capítulo 6. Presupuesto

Al tratarse de software libre el presupuesto necesario para la implementación de este proyecto es relativamente bajo, al no tener que adquirir ninguna licencia de software, así que solo deberíamos tener en cuenta los gastos normales derivados de tener un servidor de cursos en la nube, pudiendo escoger varias alternativas, ya bien sea via hosting en máquinas externas a la universidad, o montando maquinas propias que se encarguen de alojar tanto a los servidores de Open edX, como a los servidores de xAPI (LRS).

Para el cálculo de este proyecto se han tenido en cuenta los siguientes conceptos:

- Horas de inversión en el proceso de desarrollo:
 - Tiempo empleado.
 - Precio/hora Ingeniero Informático.
- Servidor en la nube para Open edX y LRS

Para llevar a cabo este proyecto se ha estimado que han sido necesarias un total de 240 horas, para las cuales se ha fijado un coste de 40 €/hora. Teniendo en cuenta esto el presupuesto del proyecto para este apartado es de 9600 €.

Adicionalmente se ha tenido en cuenta dentro de este presupuesto el coste de tener alojado en la nube de Amazon ambos servidores por razones de escalabilidad del proyecto, para ello se calcula que el coste anual sea de 639,48 € sólo en hosting de servidores.

Tipos	Descripción	Coste
Desarrollo informático	Desarrollo de las implementaciones llevadas a cabo	9600 €
Servidor en la nube	Servidor en la nube para Open edX y servidor de LRS	639,48 €

Tabla 4. Tabla resumen del presupuesto general.

Teniendo en cuenta todos estos conceptos se determina que el coste total de este proyecto asciende a un total de **10239,48 euros**, impuestos incluidos.

Bibliografía

¹Deterding S., Meaningful Play: Getting “Gamification” Right, Google Tech Talk, 2011

³Hunicke Robin, MDA: A Formal Approach to Game Design and Game Research.

Domínguez, Gamifying Learning, 2013.

⁴GLOVER, Ian. Play as you learn: gamification as a technique for motivating learners, 2013.

⁵Carina Soledad González González, A Mora Carreño, Técnicas de gamificación aplicadas en la docencia de Ingeniería Informática, 2015.

¹¹Castro Solís E. Estándares en los Sistemas de Gestión de Aprendizaje. Dirección de Tecnología Educativa IPN.

Referencias web.

⁷<http://www.edx.org>

<http://www.adlnet.gov>

<http://comunidad.iebschool.com/iebs/gamification-narrativa-transmedia/gamificacion-clientes-empleados/>

<http://www.aaai.org/Papers/Workshops/2004/WS-04-04/WS04-04-001.pdf>

<http://www.jenjenson.com/courses/learninggame/wp-content/uploads/2010/08/gamifyinglearning.pdf>

⁸ <https://captainup.com/docs/getting-started/overview>

²<http://scorm.com>

⁶Meister, J. C. (2013, January 2). How Deloitte Made Learning a Game. Retrieved from HBR Blog Network:

http://blogs.hbr.org/cs/2013/01/how_deloitte_made_learning_a_g.html

⁹<https://www.gse.upenn.edu/pressroom/press-releases/2013/12/penn-gse-study-shows-moocs-have-relatively-few-active-users-only-few-persist>

¹⁰ www.Capterra.com

Apéndices

A.1 Ejemplo Oración xAPI

```
/******  
*{  
    "id":      "12345678-1234-5678-1234-567812345678",  
              "actor":{  
                "mbox":"mailto:xapi@adlnet.gov"  
              },  
              "verb":{  
                "id":"http://adlnet.gov/expapi/verbs/created",  
                "display":{"  
                  "en-US":"created"  
                }}  
              },  
              "object":{  
                "id":"http://example.adlnet.gov/xapi/example/activity"  
              }  
}
```

*

* AUTOR:
* Advance Distributed learning

*

*

* FECHA:
* 01/10/2014

*

* DESCRIPCION

* Ejemplo de la oración más simple que podemos tener en xAPI, en ella podemos diferenciar cuatro campos principales, un identificador de la oración, un actor que es quien realiza la acción, un verbo que nos dice que acción está realizando, y un objeto que nos dice que es lo que está haciendo

*****/

A.2 Algoritmo xAPI-Captain Up

```
/******  
*  
* <div id='cptup-ready'></div>  
<script data-cfasync='false' type='text/javascript'>  
  window.captain = {up: function(fn) { captain.topics.push(fn) }, topics: []};  
  // Add your settings here:  
  captain.up({  
    api_key: '5577505dcd220c9b0a00000b',  
  user{  
    id: ${ user.id }  
  }  
  });  
</script>  
<script data-cfasync='false' type='text/javascript'>  
  (function() {  
    var cpt = document.createElement('script');  
    cpt.type = 'text/javascript'; cpt.async = true;  
    cpt.src = 'https://captainup.com/assets/embed.en.js';  
    (document.getElementsByTagName('head')[0] ||  
document.getElementsByTagName('body')[0]).appendChild(cpt);  
  })();  
</script>  
*****  
*  
* AUTOR:  
* Daniel Afonso González  
*  
* FECHA  
* 20/05/15  
* DESCRIPCIÓN  
*  
* Codigo Javascript necesario para la carga del widget de Captain Up en  
nuestro Open edX  
*****/
```

A.3 Algoritmo xAPI-Captain Up

```
/******  
*  
* def New_statement(sender,instance, created, **kwargs):  
  if (instance.verb.display.get('en-US')== 'passed'):  
  
    #ahora filtramos por actividad  
    data={  
      "app":"5577505dcd220c9b0a00000b",  
      "secret":SECRET_TOKEN,  
      "user":instance.actor.account_name,  
      "action[name]":"answer",  
      "action[data][url]":instance.object_activity.activity_id,  
      "action[entity]  
[name]":instance.object_activity.activity_definition_name.get('en-US')}}  
      r= requests.post(CAPTAIN_URL_Actions,data=data)  
  
    #Los que hayan terminado de ver un video  
    if (instance.verb.display.get('en-US')== 'Completed'):  
      data={  
        "app":"5577505dcd220c9b0a00000b",  
        "secret":SECRET_TOKEN,  
        "user":instance.actor.account_name,  
        "action[name]":"watch",  
        "action[data][url]":instance.object_activity.activity_id,  
        "action[entity]  
[name]":instance.object_activity.activity_definition_name.get('en-US')}}  
        r= requests.post(CAPTAIN_URL_Actions,data=data)  
  
*  
*****  
*  
* AUTOR:  
* Daniel Afonso González  
*
```

* FECHA

* 20/05/15

* DESCRIPCIÓN

* Algoritmo en el que primero se analiza la actividad que se acaba de registrar en el servidor y en * base a eso se formatea un petición a la API de Captain Up con la información necesaria para

* que se incluya en la base de datos de nuestro motor de gamificación

*

*****/