

UNIVERSIDAD DE



VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

**DESARROLLO DEL NUEVO MÓDULO PARA
MOODLE e-Liza v2 PARA LA
AUTOEVALUACIÓN DE CONOCIMIENTOS EN
APOYO AL GRUPO DE TELEMÁTICA E
IMAGEN**

e-Liza v2

Autor:

D. Roberto Romero Oraá

Tutora:

Dña. Miriam Antón Rodríguez

TÍTULO: **Desarrollo del nuevo módulo para Moodle e-Liza v2 para la autoevaluación de conocimientos en apoyo al Grupo de Telemática e Imagen**

AUTOR: **D. Roberto Romero Oraá**

TUTOR: **Dña. Miriam Antón Rodríguez**

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **Dña. Miriam Antón Rodríguez**

VOCAL: **D. Mario Martínez Zarzuela**

SECRETARIO: **D. David González Ortega**

SUPLENTE: **D. Francisco Javier Díaz Pernas**

SUPLENTE: **Dña. M^a Ángeles Pérez Juárez**

FECHA: **11 de Septiembre de 2014**

CALIFICACIÓN:



RESUMEN

En este TFG se pretende desarrollar una herramienta **e-Learning** que sirva de soporte al trabajo que desarrolla el Grupo de Telemática e Imagen, que pertenece al G.I.R. Sociedad de la Información de la Universidad de Valladolid.

La idea nace fruto de la necesidad de actualización, motivada por el rápido avance de las tecnologías y los nuevos planes de estudio, de una versión inicial del módulo para Moodle **e-Liza**, como juego formativo que permite la autoevaluación de conocimientos. En el contexto de los **serious games** y, tras un estudio de **CMS** orientados al aprendizaje electrónico, se revela la utilidad de este tipo de aplicaciones para los nuevos métodos de docencia. Así, particularizando el trabajo, se plantea el desarrollo de una nueva versión de e-Liza que sea compatible con las últimas versiones de Moodle y que corrija las carencias de la versión inicial, aportando mejoras funcionales y estéticas, con objeto de lograr una herramienta ágil, práctica y segura.

Para ello, se hace uso de un amplio abanico de tecnologías, librerías y plataformas para el desarrollo web (HTML5, CSS3, JavaScript, PHP, MySQL, XML, Moodle y Ajax).

PALABRAS CLAVE

E-learning, juegos formativos, CMS, Moodle, Ajax, PHP, CSS3, HTML5, JavaScript y SQL.



ABSTRACT

In this dissertation, an **e-Learning** tool has been developed to work as a support to the assignment performed by the Telematics and Image Group, which belongs to G.I.R. Information Society at the University of Valladolid.

The idea was born out of the need to renovate an initial version of the module for the so-called Moodle **e-Liza**, which is a learning game that allows the self-assessment of knowledge. It has been motivated by the fast advance of technology and new curricula. After checking and studying the e-learning-oriented **CMS**, the usefulness of this type of applications for new teaching methods is proven within the context of **serious games**. For this reason, a new version of e-Liza has been developed. It will be compatible with the latest versions of Moodle and will correct deficiencies in the initial version by providing functional and aesthetic improvements in order to get a flexible, practical and safe tool.

To do this, a wide range of technologies, libraries and platforms for web development (HTML5, CSS3, JavaScript, PHP, MySQL, XML, and Ajax Moodle) have been used.

KEYWORDS

E-learning, serious games, CMS, Moodle, Ajax, PHP, CSS3, HTML5, JavaScript y SQL.

AGRADECIMIENTOS

“A mis padres y hermanos por su plena confianza en mí.

A mis compañeros de Universidad y viejos amigos.

A Anaïs, por haberme acompañado en todo momento en este viaje.

Y a mí tutora, Miriam, por brindarme la oportunidad de realizar este proyecto.”

TABLA DE CONTENIDOS

RESUMEN.....	3
PALABRAS CLAVE.....	3
ABSTRACT	4
KEYWORDS	4
AGRADECIMIENTOS	5
TABLA DE CONTENIDOS.....	6
LISTA DE ILUSTRACIONES.....	10
LISTA DE TABLAS	12
PARTE 1 INTRODUCCIÓN Y CONOCIMIENTOS PREVIOS.....	13
Capítulo 1: Introducción	14
Introducción al proyecto.....	14
Motivación	14
Objetivos	16
Metodología.....	17
Introducción al documento	19
Capítulo 2.- Conocimientos previos.....	21
Entorno del trabajo.....	21
E-learning	21
CMS (Content Management System).....	22
LMS (Learning Management System)	25
Moodle	29
Moodle v2.....	33
Serious game	37
Capítulo 3.- Descripción del problema e hipótesis de trabajo	39
PARTE 2: DESARROLLO DEL PROYECTO: e-Liza v2	42
Capítulo 4. Aspectos técnicos	43

Estructura de ficheros del módulo	43
Pautas de programación	55
Reglas generales	55
Estilo de código.....	56
Pautas para la estructura de la base de datos.....	58
Normas de seguridad	59
Diseño de la base de datos	61
Tabla eliza	64
Tabla eliza_questions.....	65
Tabla eliza_answers	66
Tabla eliza_questions_games	67
Tabla eliza_tags.....	68
Tabla eliza_tags_questions	68
Tabla eliza_suggestions.....	69
Tabla eliza_suggestions_answers.....	70
Tabla eliza_sessions.....	71
Tabla eliza_results.....	72
Tabla eliza_group_games	73
Tabla eliza_group_games_users.....	74
Capítulo 5. Aspectos funcionales	75
Jugar	75
Administrar.....	78
Banco de preguntas del curso.....	78
Preguntas del juego	82
Preguntas propuestas por los alumnos	83
Proponer preguntas.....	84
Estadísticas	86
Estadísticas completas.....	86

Estadísticas personales.....	91
Jugar en grupo.....	92
Capítulo 6. Manual de usuario.....	94
Manual para el profesor.....	95
Instalación del módulo.....	95
Desinstalación del módulo.....	96
Creación de la actividad.....	97
Menú principal.....	98
Administrar.....	98
Ver estadísticas completas.....	105
Manual para el alumno.....	106
Menú principal.....	106
Jugar.....	106
Jugar en grupo.....	108
Proponer preguntas.....	109
Ver estadísticas personales.....	111
PARTE 3: FINALIZACIÓN.....	112
Capítulo 7. Presupuesto económico.....	113
Capítulo 8. Conclusiones.....	116
Capítulo 9. Líneas de futuro.....	119
BIBLIOGRAFÍA.....	121
ANEXOS.....	125
Anexo 1. Roles estándar de Moodle.....	126
Anexo 2. PHP.....	127
Anexo 3. AJAX.....	131
Anexo 4. CSS3.....	133
Anexo 5. JavaScript.....	138
Anexo 6. MySQL y SQL.....	140



Anexo 7. Código de generación de los datos de la gráfica temporal.	142
Anexo 8. Código de generación de la lista de usuarios matriculados en el curso.	144
Anexo 9. Diagrama de casos de uso.....	146
Anexo 10. Diagramas de actividades UML.....	147
Jugar	147
Jugar en grupo	148
Administrar.....	149
Proponer preguntas.....	150
Estadísticas	150

LISTA DE ILUSTRACIONES

Ilustración 1: Captura del juego e-Liza v1	41
Ilustración 2: Iconos de actividad para Moodle	49
Ilustración 3: Logo e-Liza v2.....	49
Ilustración 4: Menú principal para el rol profesor	51
Ilustración 5: Menú principal para el rol alumno	51
Ilustración 6: Gráficas generadas con pChart	53
Ilustración 7: Esquema de la base de datos	63
Ilustración 8: Secciones del programa.....	75
Ilustración 9: Gráfica circular o de tarta	88
Ilustración 10: Gráfica temporal.....	88
Ilustración 11: Instalación mediante archivo ZIP.....	95
Ilustración 12: Instalación manual en el servidor	96
Ilustración 13: Desinstalación del módulo.....	97
Ilustración 14: Creación de la actividad.....	97
Ilustración 15: Menú principal del programa para el profesor.....	98
Ilustración 16: Configuración general del juego	99
Ilustración 17: buscador del banco de preguntas.....	100
Ilustración 18: Nueva pregunta	100
Ilustración 19: Nueva respuesta.....	101
Ilustración 20: Nueva etiqueta.....	103
Ilustración 21: Flechas de ordenación de las preguntas	104
Ilustración 22: Ejemplo de estados para preguntas propuestas.....	105
Ilustración 23: Menú principal del programa para el alumno.....	106
Ilustración 24: Panel de información del juego	107
Ilustración 25: Valor de la pregunta y porcentaje de penalización	108
Ilustración 26: Ejemplo de gestión de preguntas propuestas personales	109
Ilustración 27: Panel de Nueva pregunta	110
Ilustración 28: Panel de Nueva respuesta.....	111
Ilustración 29: Evolución del uso de PHP	128
Ilustración 30: Uso de PHP según versiones.....	128
Ilustración 31: Gráfica comparativa de diferentes lenguajes de programación del lado del servidor.....	129



Ilustración 32: Relación de la extensión de distintos lenguajes de programación con su tráfico	129
Ilustración 33: Proceso de funcionamiento de AJAX.....	131
Ilustración 34: CSS3. Propiedades: border-radius y shadows.....	136
Ilustración 35: CSS3. Media queries.....	137
Ilustración 36. Uso de los principales lenguajes de programación del lado del cliente	139
Ilustración 37. Relación Bases de datos y Gestores de Bases de Datos .	140
Ilustración 38: Diagrama de casos de uso.....	146
Ilustración 39: Diagrama de actividad: Jugar	147
Ilustración 40: Diagrama de actividad: Jugar en grupo	148
Ilustración 41: Diagrama de actividad: Administrar	149
Ilustración 42: Diagrama de actividad: Proponer preguntas.....	150
Ilustración 43: Diagrama de actividad: Estadísticas	151

LISTA DE TABLAS

Tabla 1: eliza	65
Tabla 2: eliza_questions.....	66
Tabla 3: eliza_answers	67
Tabla 4: eliza_tags.....	68
Tabla 5: eliza_tags_questions	69
Tabla 6: eliza_suggestions	70
Tabla 7: eliza_suggestions_answers.....	71
Tabla 8: eliza_sessions.....	72
Tabla 9: eliza_results.....	73
Tabla 10: eliza_group_games	74
Tabla 11: eliza_group_games_users.....	74

PARTE 1

INTRODUCCIÓN Y CONOCIMIENTOS PREVIOS



Capítulo 1: Introducción

Introducción al proyecto

Motivación

En las últimas décadas se ha ido imponiendo la idea de concebir el aprendizaje como el resultado de la elaboración de conocimientos a partir del establecimiento de la relación entre unos conocimientos previos y los nuevos que se desean adquirir.

Para que ello sea posible el rol del docente ha de ir evolucionando en el sentido de facilitar entornos en los que puedan tener lugar esas interacciones entre los conocimientos que ya tienen los alumnos y los nuevos que desean adquirir. De ahí la necesidad de buscar **nuevas herramientas** y **métodos de enseñanza** que colaboren al propósito de mejorar el proceso de aprendizaje.

En el ámbito universitario, los métodos de enseñanza evolucionan en paralelo a la mejora experimentada por las **nuevas tecnologías** en la docencia. Cuando en el año 2008 se celebró la Conferencia de Rectores de las Universidades Españolas para tratar el tema de las TIC's, se reveló que prácticamente la totalidad de las universidades españolas poseían un plan institucional de **docencia virtual**, bien ya implantado (78%) o bien en desarrollo (18%). Además, la apuesta definitiva por la docencia virtual, que en 2006 sólo cubría el 67% de posibles iniciativas, se elevó progresivamente hasta alcanzar el 79% en 2010. [12].

En este sentido, son continuos los esfuerzos realizados por incorporar las Tecnologías de la Información y Comunicaciones (TIC's) a la docencia universitaria en congruencia con el ambicioso plan que han puesto en marcha los países del viejo continente para favorecer, en materia de educación, la convergencia europea: el "Espacio Europeo de Educación Superior" (EEES), estando entre sus objetivos el de impulsar un cambio en las metodologías docentes al tiempo que impulsar a las universidades a aprovechar su capacidad de **innovación** [8].



Con este contexto, Juan Manuel Alonso Gutiérrez, en su Proyecto de Fin de Carrera de la Universidad de Valladolid, incorporó en el año 2008 el desarrollo de una herramienta de **autoevaluación de conocimientos basado en juegos**. La herramienta estaba basada en el popular juego de la TV americana **Jeopardy!**, y consistía en facilitar a los alumnos una mejor comprensión de la asignatura mediante un juego de preguntas con diferentes categorías y de distinto valor. El alumno podría entrenar en su casa, individualmente o en grupo, y habría una partida oficial de carácter presencial, realizada en el laboratorio, y que también podría desarrollarse individualmente o en grupo. El objetivo del juego sería finalizar con la mayor cantidad de puntos en el menor tiempo posible. [6]

Este prototipo (*Jeopardi*) desarrollado para la plataforma Moodle acabó evolucionando en 2009, hacia un nuevo juego que adoptó el nombre de **e-Liza** basado en las tecnologías AJAX y Comet, gracias al Proyecto de Fin de carrera de Sara San Martín del Pozo. [9]

La aplicación de este nuevo modelo en un escenario real, sacó a la luz una serie de **carencias** que se pretendieron subsanar con la realización de un **nuevo proyecto** de Fin de Carrera por Alfonso Ruifernández Pérez en 2010. Entre ellas destacaba el autor la no diferenciación de usuarios, problemas con las estadísticas, problemas con las hojas de estilo al actualizar a la versión Moodle 1.9, errores de diseño en los diferentes exploradores y un largo etcétera de pequeños detalles que hacían la aplicación no apta para su uso inminente. [10]

No solo se solucionaron algunos errores, sino que también se añadieron una serie de nuevas funcionalidades que hicieron la plataforma más accesible y cómoda de usar tanto para el alumno como para el profesor. Pero aun así, también en este nuevo modelo se han ido detectando **problemas**, sobre todo de seguridad, en cuanto a que pueden verse violadas algunas reglas del juego y, fundamentalmente de falta de comodidad y agilidad en su uso global, especialmente para el profesor, en la gestión del juego.

Por eso se hace necesaria una **nueva versión** que corrija las deficiencias comentadas y porque, además, el soporte Moodle que lo sostiene también ha evolucionado de manera notoria.



Objetivos

Ha de ser objetivo prioritario del proyecto dar respuesta a la necesidad de **actualizar** el sitio web multifuncional que sirva de apoyo tanto a la investigación realizada por el Grupo de Telemática e Imagen, como a la docencia. En este sentido, la constante actualización de la plataforma virtual educativa impuesta por la Universidad de Valladolid, Moodle, exige a su vez renovar los módulos utilizados para asegurar su correcto funcionamiento en sus últimas versiones.

Las necesidades de actualización parten no sólo del rápido avance de las tecnologías web y de la mejora significativa que produce el uso de gestores de contenidos con el paso del tiempo, sino también de las distintas necesidades en el apoyo a la docencia, que se dan con los nuevos planes de estudios.

Otros objetivos generales que pretende la realización del Proyecto son:

- Poner en práctica el conjunto de **competencias** asociadas al Grado y demostrar que se han adquirido suficientemente dichas competencias.
- Desarrollar **metodologías** y **destrezas** de aprendizaje autónomo eficiente para la adaptación y actualización de nuevos conocimientos y avances científicos.
- Ponerse al día en las mejores **soluciones** y los últimos avances en el tema.
- Llevar a la práctica la capacidad de **organización, planificación y gestión** del tiempo necesarios para poder llevar a buen término semejante Proyecto.

Y en el contexto arriba explicado, serían objetivos específicos del Proyecto los siguientes:

- **Solventar** las **carencias** de las versiones anteriores del mismo módulo.
- **Satisfacer** las **exigencias** técnicas, estéticas y de seguridad requeridas.



- Y, como objetivo último, se pretende conseguir una **aplicación** que pueda ser considerada de gran utilidad en la docencia y que sirva de herramienta totalmente eficaz para la autoevaluación de los conocimientos del alumno.

Metodología

Se plantea un **plan de trabajo** definido y apoyado en todo momento en la consideración de antiguos trabajos relacionados, en el uso de fuentes bibliográficas fiables y en la observancia a las pautas de señaladas por la tutora del Proyecto.

Así, en primer lugar, se inicia un proceso de **documentación** donde se estudian el contexto del proyecto, los conceptos necesarios para su desarrollo y la viabilidad de cada nueva idea surgida, así como su forma de implementarla. Esto implica una extensa búsqueda de información y el planteamiento de una sólida hipótesis de trabajo, sin despreciar en ningún momento los objetivos marcados.

En este punto, se recopilarán una serie de aspectos tanto técnicos como funcionales, donde el diseño de algunos diagramas ayudará a enfocar los requerimientos pertinentes y clarificar la forma en que se va a desenvolver la labor. Esto constituye el siguiente paso, el proceso de **diseño**.

En tercer lugar, se lleva a cabo la puesta a punto del **entorno de trabajo**. Es primordial hacerse con las herramientas necesarias que permitan elaborar el proyecto de la mejor forma posible, con una buena organización y simplicidad. El proceso de programación, como principal tarea, se desarrollará con ayuda de la herramienta Adobe **Dreamweaver** [REF], orientada a la construcción, diseño y edición de páginas web facilitando, extensiblemente, la escritura de código en los distintos lenguajes que se van a utilizar. Aunque no se trata precisamente de una aplicación liviana, minimalista y gratuita, cumple amablemente con los requisitos óptimos de cualquier editor de texto orientado a la programación, entre los que cabe destacar la detección de errores en el código a tiempo real y las sugerencias para el código



en forma de lista de candidatos que completarían una entrada de texto automáticamente. Además de la capacidad de agregar extensiones y su integración con otras herramientas, la principal característica de Dreamwwaver es su editor WYSIWYG (What You See Is What You Get), que permite diseñar la aplicación web de forma gráfica sin escribir código alguno, el cual será generado automáticamente. No obstante esta propiedad no ha sido utilizada porque ofrece como resultado un estilo de código poco limpio y normalmente más extenso de lo necesario. Una alternativa interesante a este editor de texto, entre otras muchas, es la herramienta **Notepad++** [REF], gratuita y mucho más ligera que la anterior. Para completar el entorno de trabajo se atiende a las necesidades de las sucesivas partes de la metodología.

La siguiente etapa del proyecto abarca el **desarrollo** en sí de la aplicación, que se dará simultáneamente a la fase de **depuración y pruebas**. Siguiendo las pautas técnicas y recomendaciones estudiadas anteriormente se organiza una estructura de ficheros que se irán creando al mismo que tiempo que se garantiza su funcionamiento. Para ejecutar los resultados se precisa de una instalación de la plataforma educativa con la que se trabaja sobre un servidor web que interprete el lenguaje de script PHP e incluya una base de datos. **XAMP** [REF] es la herramienta elegida para este propósito, permitiendo aunar el servidor web Apache, un intérprete de PHP y una base de datos MySQL en un mismo paquete, de forma que se disponga de un servidor local donde ir testeando el funcionamiento de la aplicación a desarrollar. Distintos navegadores web harán las veces de cliente que hará uso de ese servidor para ejecutar las distintas partes del programa.

En última instancia del trabajo, es inexcusable comprobar si opera o no como se esperaba. Habrá que llevar a cabo una serie de **pruebas globales** poniéndose en la piel de los diferentes roles para los que está diseñado y obtener unas conclusiones concretas de todo este proyecto y del resultado final obtenido.

Introducción al documento

El documento se estructura en tres partes desarrolladas en nueve capítulos cuyos contenidos se describen a continuación y en una serie de anexos que refuerzan la explicación:

Parte 1: Introducción y conocimientos previos

Capítulo 1.- Introducción.

Se introduce el tema abordado en el proyecto describiendo la trayectoria seguida en el ámbito del e-learning y los serious games en el seno de la UVA, la necesidad a la que se quiere dar respuesta y los beneficios que podrían aportar. Y, al final, se plantea esta Introducción al documento, donde se explica la estructura del mismo, con el contenido que se verá en cada capítulo.

Capítulo 2.- Conocimientos previos.

Se describen distintos conceptos concernientes al contexto (e-Learning, CMS, LMS, etc.) y finalmente se analizan las principales plataformas e-Learning de código abierto disponibles en la actualidad haciendo un repaso de las (sus) tecnologías asociadas. disponibles para el desarrollo de la aplicación planteada y justificando la elección realizada para trabajar. Así, se hace una primera referencia a Moodle, sus características básicas y por qué se utiliza en este caso.

Capítulo 3.- Descripción del problema e hipótesis de trabajo.

Se analiza la última versión estable del módulo e-liza identificando sus principales problemas y planteando la forma en que van a solventarse. La hipótesis de trabajo puede entenderse como una "conjetura plausible", con objeto de descubrir la respuesta más satisfactoria que se pueda proponer, habida cuenta de los vacíos detectados y del estado de los conocimientos previos estudiados.

Parte 2: Desarrollo del proyecto: e-Liza v2

Capítulo 4.- e-liza v.2: Aspectos técnicos.

Se estudian pautas concretas y recomendaciones para el desarrollo cualquier módulo para Moodle. Asimismo se define la

estructura de ficheros de la nueva versión de e-Liza y el nuevo esquema de la base de datos.

Capítulo 5.- e-liza v.2: Aspectos funcionales.

Se analizan las distintas funcionalidades de e-Liza v2, para cada uno de los roles potenciales, con algunos comentarios relativos a su implementación

Capítulo 6.- e-liza v.2: Manual de usuario

Presenta una guía para que los posibles usuarios del sistema no tengan dificultades para sacarle el máximo rendimiento a la aplicación.

Parte 3: Finalización

Capítulo 7.- Presupuesto económico

Expone un estudio del coste de la aplicación, dando precio aproximado al desarrollo de la herramienta teniendo en cuenta diversos factores.

Capítulo 8.- Conclusiones

Aquí se valora el grado de consecución de los objetivos planteados al principio así como las competencias demostradas para el desarrollo del trabajo.

Capítulo 9.- Líneas de futuro

Se deja abierta la proyección de futuro de e-Liza v2 avanzando posibles líneas de mejora.

Anexos

Se hace un repaso de los roles que están disponibles por defecto en Moodle, se detallan **características de las tecnologías** en las que se basa la plataforma educativa (CSS3, PHP, JavaScript, etc), se presenta algún fragmento de **código de programación** interesante y se reúnen todos los **diagramas** elaborados con objeto de completar el esclarecimiento del funcionamiento de la aplicación.

Capítulo 2.- Conocimientos previos

Entorno del trabajo

E-learning

Una de las herramientas más avanzadas para mejorar el complejo proceso de aprendizaje es el aprendizaje electrónico, el cual fomenta el *e-Learning* como medio de enseñanza flexible y a distancia.

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 mediante la utilización de diversas herramientas informáticas. [1]

Este nuevo concepto educativo es una revolucionaria modalidad de capacitación que posibilitó Internet, y que hoy se posiciona como la forma de capacitación predominante en el futuro. Este sistema ha transformado la educación, abriendo puertas al aprendizaje individual y organizacional. Por eso, hoy en día está ocupando un lugar cada vez más destacado y reconocido dentro de las organizaciones empresariales y educativas.

Los beneficios del e-learning son muchos pero destacan:

- 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 conexión a Internet, cuando les surge la necesidad.
- Flexibilidad de la agenda: no se requiere que un grupo de personas coincidan en tiempo y espacio.

El término "e-learning" es la simplificación de Electronic Learning que reúne a las diferentes tecnologías y a los aspectos pedagógicos de la



enseñanza y el aprendizaje. De hecho, 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 Tecnología de la Información y la Comunicación, mediante la selección, diseño, personalización, implementación, alojamiento y mantenimiento de soluciones en dónde se integran tecnologías propietarias y de código abierto (Open Source).

Los **aspectos pedagógicos** son como el alma del e-learning y van a trabajar sobre los contenidos. Puede que al principio sean los menos tangibles, pero serán, al final, los componentes más relevantes en términos de eficacia de los objetivos de enseñanza y aprendizaje fijados.

Sin embargo, serán los **aspectos tecnológicos** los que centren nuestro interés. Especialmente los que se refieren a la gestión de los contenidos. Y para ello, contamos con los sistemas de gestión de contenidos, **CMS** (Content Management System),

CMS (Content Management System)

Los **CMS** son programas que permiten crear una estructura de soporte para la creación y administración de contenidos (creación, clasificación, publicación, actualización, eliminación o archivo de todos los contenidos) sin la exigencia de grandes conocimientos técnicos por parte del usuario. Surgieron tanto para facilitar el desarrollo de sitios Web dinámicos como la gestión de elevados volúmenes de información y contenidos en dichos sitios Web.

Los CMS son aplicaciones **modulares** que se apoyan en bases de datos o repositorios y están formadas por dos elementos: la Aplicación Gestora de Contenidos (**CMA**) y la Aplicación de Presentación de Contenidos (**CDA**). Estos elementos se utilizan mediante una única interfaz más o menos intuitiva, dependiendo de la aplicación, y por lo



general de tipo Web. El CMA, permite al autor del sitio realizar la creación, actualización y eliminación del contenido del sitio Web, sin necesidad de que éste posea conocimientos de *HTML* u otro lenguaje. El CDA por su parte, permite al autor elegir, mediante la aplicación de plantillas, el formato en que se representarán los contenidos [6].

El nacimiento del denominado **Open Source** (código abierto), software desarrollado y distribuido de forma libre, ha provocado una extraordinaria proliferación de CMS con la más variada orientación. Una muestra de ello se refiere a continuación [4]:

- **Extpattern.** Es un CMS open source, sencillo y flexible escrito en PHP y que usa mySQL para la base de datos.
- **Drupal.** Es un sistema de gestión de contenidos que además de las funcionalidades básicas permite tener foros, Newsletters, Podcasting y galerías de imágenes. Drupal está escrito en PHP y SQL.
- **WordPress.** Es el sistema CMS Open Source más usado gracias a su versatilidad (valido para plataformas webs, blogs..), configuración avanzada orientada al SEO y a la ayuda de numerosos plugins con el que completar su funcionalidad.
- **Joomla.** Uno de los CMS más reconocidos, nos permite crear sitios web y aplicaciones online muy poderosas. Dispone de una interesante comunidad de usuarios.
- **Nucleus.** Escrito en PHP y mySQL este software nos da la posibilidad de manejar múltiples blogs para múltiples autores.
- **dotCMS.** Otro software premiado y construido en Java que incluye funcionalidades como soporte para hosting virtual, contenido estructurado, clustering y puede funcionar en múltiples bases de datos. Incluye RSS feeds, calendario AJAX, noticias, blogs, foros, motor de búsqueda incorporado,...
- **Alfresco.** CMS OpenSource para sistemas Windows y Unix. Está diseñado para usuarios que necesitan muchas funciones y escalabilidad. Alfresco está escrito en Java y posee integración de escritorio y administración completa de documentos.



- **Plone**. Sistema de gestión de contenidos escrito en Python. Todas sus funcionalidades son personalizables y dispone de extensiones gratuitas disponibles.
- **e107**. CMS construido con PHP y soporte de base de datos via MySQL que posee noticias, RSS feed, sistema de template y foro.
- **eZ Publish**. Posee una interesante función para importar/exportar documentos de texto. Incluso puede traducir tu contenido a otros lenguajes. Incluye galería de imágenes, publicación de videos y un bonito control de administración de usuario.
- **Mambo**. Es un sistema de gestión de contenidos orientado a los negocios muy fácil de utilizar. Es completamente personalizable y dispone de una activa comunidad de usuarios donde buscar ayuda. Incluye noticias, secciones de servicios o productos completamente editables, editor de contenido WYSIWYG y administración de publicidades.
- **Diferior**. Es un CMS multipropósito que nos provee de todas las herramientas necesarias para publicar y organizar nuestro contenido. Es capaz de distribuir archives vía protocolos BitTorrent, HTTP y FTP, posee foros incorporados y sistema de blogs.
- **Symphony**. Es un sistema de gestión de contenidos hecho para desarrolladores web, utiliza XSLT para proveer de gran flexibilidad.
- **Frog CMS**. Es un CMS simplificado que utiliza PHP 5 y MySQL 4. Posee las capacidades de arrastrar y soltar, contenido de páginas flexible y menú de navegación muy personalizable. Además se lo puede personalizar con distintos temas y plugins.
- **xpressionEngine**. Es un CMS flexible y fácil de utilizar que requiere PHP 4+ y Mysql 3+. Posee una gran función de cache incorporada e incluso dispone de un carrito de compra.

Todos los CMS son útiles para administrar y gestionar contenidos de cualquier ámbito. Pero a la hora de administrar y gestionar contenidos formativos, carecen de herramientas de comunicación específicas para la actividad docente, ya que su finalidad está más orientada a la gestión de contenidos en sí, más que a la gestión del aprendizaje. Por eso, se hace necesario un software que automatice la administración de acciones de formación, un Sistema de Gestión de Aprendizaje o **LMS** (Learning Management System).



LMS (*Learning Management System*)

Podríamos decir que un **LMS** puede considerarse como un CMS de propósito específico, concretamente educativo, que potencia las posibilidades de colaboración e interactividad que puede ofrecer un espacio virtual, siempre que se empleen los recursos adecuados para tal finalidad. Los LMS, por tanto, se adecuan a los propósitos específicos del denominado eLearning (aprendizaje en línea).

Son variadas las funcionalidades de un LMS: registra a todos los actores que intervienen en el acto de aprendizaje (alumnos, profesores, administradores, etc.), organiza los diferentes cursos en un catálogo, almacena datos sobre los usuarios, realiza un seguimiento del aprendizaje y la temporización de los trámites y genera informes automáticamente para tareas de gestión específicas. También desarrolla procesos de comunicación, e incluso algunos LMS permiten posibilidades de autoría de contenidos. Éstos serían los que se conocen como Sistemas de Gestión de Contenidos de Aprendizaje **LCMS** (Learning Content Management Systems, en inglés). [6]

Un LMS y un LCMS, aunque complementarios, son dos sistemas muy diferentes que sirven para distintos propósitos. Podemos decir que un LMS es un software que planifica y gestiona los eventos de aprendizaje dentro de una organización, incluyendo el aula *online* o virtual y los cursos dirigidos por un instructor. Por el contrario, un LCMS es un software para la gestión de contenidos de aprendizaje de los diversos programas de capacitación que se configuran en el desarrollo en toda la organización. En este segundo caso, se proporciona a los desarrolladores, autores, diseñadores instruccionales y expertos en la materia los medios para crear y reutilizar el contenido de aprendizaje y reducir la duplicación de los esfuerzos de desarrollo, ya que un LCMS crea, almacena, ensambla y entrega de forma personalizada el contenido en forma de objetos de aprendizaje específicos. Un LMS nos ayuda a gestionar la administración de una enseñanza o de un curso en todos sus aspectos: gestionar usuarios, recursos, actividades, módulos, permisos, generar informes, evaluaciones, calificaciones, comunicación de foros, videoconferencias, chats y demás. Un LCMS es lo mismo que



un LMS pero con una particularidad, que es la propia de los CMS: el hecho de poder administrar todos los contenidos del sistema. [5]

Evidentemente, en el presente trabajo nuestro interés se ha de centrar en los de tipo e-learning. Y, a mayores en los de software desarrollado y distribuido de forma libre. Entre ellos caben diferentes alternativas de las que destacamos las siguientes: [11]

- **Atutor**

Este proyecto empezó en 2002 en colaboración con el *Adaptive Technology Resource Centre (ATRC)* de la *Toronto University* [7]. Este centro es un líder internacionalmente reconocido en el desarrollo de tecnologías y estándares que permitan a las personas con discapacidades el acceso a las oportunidades *e-learning*. Utiliza las tecnologías PHP, Apache y MySQL, y soporta todos los sistemas operativos en los cuales es posible instalar PHP, Apache y My SQL (ej. Windows, Linux, UNix, etc). Entre sus características principales están el soporte de estándares, facilidades de importación/exportación de contenidos y usuarios, herramientas de seguimiento, posibilidad de gestionar los contenidos de los cursos, es fácil crear contenidos y tests dentro de la plataforma e importar paquetes SCORM 1.2 e IMS. No obstante adolece de que sólo existe documentación en inglés y de que faltan herramientas síncronas.

- **Bazaar**

Bazaar es un desarrollo de la mayor universidad abierta de Canadá, la Universidad de Athabasca. Empezó como un sistema de "web board conferencing" pero ha evolucionado rápidamente en un sistema integrado de información. Bazaar es un sistema muy flexible y configurable y puede ser usado para distribuir cursos, portales o cualquier otro tipo de proyectos basados en la web. Utiliza las tecnología Apache, Perl y MySQL, y soporta los sistemas Operativos Linux y FreeBSD. Su principal



característica es que es muy configurable. Gracias a la creación de plantillas, casi cualquier estrategia de distribución de información es posible permitiendo gestionar los contenidos de los cursos de forma ágil. Pero el sistema no está traducido a ninguna lengua aparte del inglés y no soporta ningún E-Learning estándar.

- **Claroline**

Claroline es uno de los LMS más usados en el mundo. Muchas universidades aprecian su ambiente de aprendizaje colaborativo que permite a los enseñantes y a las instituciones educativas crear y administrar cursos en la web. Las herramientas que ofrece el sistema son muchas (gestión de los grupos, fórum, repositorios de documentos, chat, administración del perfil de los usuarios, entre otras) y dan a los usuarios la posibilidad de establecer cualquier escenario deseado. Utiliza tecnología PHP, Apache, MySQL y soporta todos los Sistemas Operativos en los que se pueda instalar PHP, Apache y MySQL (Windows, Linux, UNix, etc). Todas sus herramientas son fáciles de entender y utilizar, la navegación es intuitiva y la administración completa. Entre las nuevas funcionalidades de la última versión se halla Wiki y la posibilidad de construir un área de trabajo donde los estudiantes pueden compartir materiales de aprendizaje y ejercicios. Se pueden crear y gestionar cursos desde dentro de Claroline y es posible crear contenidos (páginas) y tests. Además soporta perfectamente el estándar de E-Learning SCORM 1.2, aunque no SCORM *runtime environment*.

- **Dokeos**

El proyecto empezó desde una versión previa de Claroline y se ha convertido en un producto por sí mismo. El objetivo es ayudar al docente a crear contenido pedagógico, a estructurar las actividades en caminos de aprendizaje, a interaccionar con los estudiantes y a seguir su evolución



mediante un sistema de informes. Utiliza tecnología PHP, Apache, MySQL y soporta todos los sistemas Operativos en los que se pueda instalar PHP, Apache y MySQL (Windows, Linux, UNix, etc). Es como una versión mejorada de Claroline siendo compatible tanto con SCORM 1.2 como con SCORM runtime environment. También ofrece tecnología plugin para la extensión de la plataforma. es una plataforma muy interesante, pero la documentación no está completa y existe solo en inglés.

- **Ganesha**

Ganesha permite al formador o a la organización formadora ofrecer a uno o varios grupos de alumnos uno o varios módulos de aprendizaje, con contenidos, addons, encuestas y tests de evaluación. También ofrece herramientas de colaboración (webmail, foro, chat, compartición de documentos) y herramientas para la tutoría en línea. Utilizada tecnología PHP, Apache, MySQL y soporta todos los sistemas operativos en los cuales es posible instalar PHP, Apache y My SQL (ej. Windows, Linux, UNix, etc). Ofrece un amplio abanico de herramientas colaborativas y una pedagogía bien planteada con la posibilidad de gestionar los contenidos de los cursos y también de subir materiales creados offline. Soporta los Estándares de E-Learning: SCORM y AICC, pero su Interfaz no es muy intuitiva.

- **Ilias**

Es una plataforma que ofrece muchas funcionalidades a todos los niveles. Con Ilias es posible establecer diferentes escenarios e incluso entornos complejos para todos los usuarios. Utiliza tecnología PHP, Apache y MySQL, y soporta todos los sistemas operativos en los cuales sea posible instalar PHP, Apache y MySQL (Windows, Linux, Unix, etc.). Es uno de los LMS más completos y potentes que existen en el



mundo del software libre. Es el más diferente en su manejo, algunas funcionalidades no son fáciles de entender y la preinstalación es un poco complicada, pero puede contar con muchas herramientas que se pueden usar e integrarse de la forma que se quiera. Ofrece la posibilidad de crear y gestionar los contenidos de los cursos e incluso una única plataforma puede soportar múltiples clientes que administren múltiples bases de datos.

- **Moodle**

Moodle es uno de los LMS más populares y está actualmente viviendo una fase explosiva de expansión. Su comunidad de usuarios y desarrolladores es muy numerosa y se caracteriza por su entusiasmo respecto al sistema. Utiliza la tecnología PHP y soporta los Sistemas Operativos de Unix, Linux, Windows, Mac OS X, Netware y cualquier otro sistema que soporte PHP. Permite crear y gestionar contenidos de cursos y tests con un sistema es muy intuitivo y fácil de usar, está traducido a más de 40 lenguas y se apoya en una gran comunidad de usuarios y desarrolladores. Los Estándares de E-Learning que soporta son SCORM 1.2 y IMS-QTI.

Y de todos ellos, sin duda el de mayor éxito en el momento actual por su facilidad de uso es la plataforma **Moodle**, en su origen acrónimo de Module Object-Oriented Dynamic Learning Environment (Entorno Modular de Aprendizaje Dinámico Orientado a Objetivos).

Además, al ser el gestor de contenidos por excelencia, extendido e impuesto en la Universidad de Valladolid, centraremos nuestro estudio en la plataforma Moodle para la cual será desarrollada la herramienta formativa.

Moodle

Moodle [7] fue creado por Martin Dougiamas, de la Universidad Tecnológica de Curtin, Australia, que basó su diseño en las ideas del



constructivismo en pedagogía, teoría según la cual el conocimiento se ha de ir construyendo en la mente del estudiante, no siendo igual de eficaz el conocimiento adquirido a través de libros o enseñanzas magistrales. La misión del profesor que operase desde el punto de vista constructivista sería la de crear un entorno centrado en el estudiante que le ayudara a construir ese conocimiento con base en sus habilidades y conocimientos propios.

En 1999 empezó probando algunos prototipos tempranos de un nuevo Learning Management System (**LMS**), las experiencias que formaron la base para su trabajo *Improving the Effectiveness of online Learning* (Mejorando la efectividad del aprendizaje a distancia). El primer sitio que existió de Moodle era de Peter Taylor en <http://smec2001.moodle.com>, en la misma Universidad Curtin, con Martin realizando la primera publicación en su propio sitio en el dominio Moodle.com en noviembre del año 2001. Ambos publicaron *An Interpretive analysis of an internet based course constructed using a new courseware tool called Moodle* (Un análisis interpretativo de un curso basado en Internet construido usando una nueva herramienta para cursos denominada Moodle).

Moodle 1.0 fue liberado en agosto de 2002. Los usuarios estaban usando Moodle en un nuevo foro, traduciendo Moodle a diferentes idiomas y creando los temas. Un año después, el primer módulo de contribución (taller) fue liberado y Moodle.org se convirtió en el brazo de la comunidad de Moodle; con Moodle.com representado el aspecto comercial.

Moodle creció rápido: el primer debate acerca de Moodle se realizó en Oxford en el año 2004 y las compañías empezaron a realizar solicitudes para ser Moodle partners (socios de Moodle).

Para comprender su éxito, hay que reconocer que Moodle era un sistema de gestión del aprendizaje (learning management system = LMS), también llamado Ambiente Virtual de Aprendizaje (VLE) para cuyo manejo bastaba con tener las habilidades básicas para navegar en Internet. Para instalarlo se necesitaba un poco más de conocimiento, pero una guía oficial lo facilitaba mucho. Incluso había previsiones en



Moodle para personas con discapacidades, soportando muchas tecnologías asistivas, tales como lectores de pantalla, aumentadores de pantalla, uso alternativo de ratón y teclas, deshabilitación de AJAX y Javascript, y más.

Moodle es un sistema para el Manejo del Aprendizaje en línea gratuito, que les permite a los educadores la creación de sus propios sitios web privados, llenos de cursos dinámicos que extienden el aprendizaje, en cualquier momento, en cualquier sitio. El núcleo de Moodle, extremadamente personalizable, viene con muchas **características** estándar:

- **Interfaz moderna**, fácil de usar. Diseñada para ser responsiva y accesible, la interfaz de Moodle es fácil de navegar, tanto en computadoras de escritorio como en dispositivos móviles.
- **Tablero Personalizado**. Permite organizar y mostrar cursos en la forma que se desee, y ver en conjunto los mensajes y tareas actuales.
- Actividades y **herramientas colaborativas**. Facilita el trabajo y el aprendizaje colectivo a través de foros, wikis, glosarios, actividades de base de datos y mucho más.
- **Calendario todo-en-uno**. La herramienta del calendario de Moodle ayuda a mantener al día el calendario académico, recordando fechas de entrega dentro del curso, reuniones grupales y otros eventos personales.
- **Gestión conveniente de archivos**. Basta arrastrar y colocar los archivos desde servicios de almacenamiento en la nube, incluyendo MS Skydrive, Dropbox y Google Drive.
- **Editor de texto** simple e intuitivo. Se puede dar el formato más conveniente al texto y añadir multimedia e imágenes con un editor que funciona con todos los navegadores de Internet y en todos los dispositivos.
- **Notificaciones**. Cuando se habilitan, los usuarios pueden recibir alertas automáticas acerca de nuevas tareas y fechas para entregarlas, publicaciones en foros y también pueden mandarse mensajes privados.



- **Monitoreo del progreso.** Los educadores y los educandos pueden monitorear el progreso y el grado de finalización con un conjunto de opciones para monitoreo de actividades individuales o recursos, y también a nivel del curso.

Desde el punto de vista de la **Administración del sistema**, destacan:

- la posibilidad de introducir un diseño personalizable del sitio.
- Autenticación e inscripciones (matriculaciones) masivas seguras.
- Capacidad Multilingüe
- Creación masiva de cursos y fácil respaldo
- Gestión del acceso de los diferentes usuarios.
- Soporta estándares abiertos.
- Alta interoperabilidad.
- Gestión simple de plugins y complementos.
- Actualizaciones regulares de seguridad
- Reportes y bitácoras detalladas

Desde el punto de vista de la **Gestión y Desarrollo de los cursos**, destacan:

- Rutas directas de aprendizaje
- Fomento de la colaboración
- Alojamiento de recursos externos
- Integración Multimedia
- Gestión de grupo
- Flujograma de puntuación
- Calificación en-línea
- Evaluación propia y por pares
- Insignias integradas
- Competencias y rúbricas
- Seguridad y privacidad

Con documentación mejorada y nueva certificación, Moodle se ha establecido desde 2007 como un líder y ganador de reconocimientos en el estándar de código abierto LMS. De 1000 sitios registrados en 2004,



ha llegado a medio millón de usuarios en 2008 y a cerca de un millón de usuarios en 2010, con cerca de 50 socios de Moodle.

Moodle v2

El tan largamente esperado **Moodle 2.0** llegó en noviembre de 2010 y ahora, se producen actualizaciones regulares prácticamente cada dos meses. La última es de julio de 2014 y está prevista la próxima para el 22 de septiembre del mismo 2014. El enfoque actual se encuentra en la tecnología móvil: una app oficial para HTML5 fue liberada en 2013 y la versión de Moodle 2.5, *Nuevas_características_de_Moodle_2.5*, incluye un tema personalizable para todos los tamaños de pantalla de dispositivos.

La inauguración de la conferencia de investigación de Moodle en 2012 sirvió como un recordatorio de que, por más avanzada que esté la tecnología, el diseño y desarrollo de Moodle está guiado por la pedagogía del construccionismo social. En septiembre del 2013, el curso Moodle MOOC oficial , Learn Moodle, (MOOC = Massive Open Online Course) introdujo a los más de 9000 participantes a las características básicas de Moodle. A los educadores de todo el mundo se les alienta para que compartan sus experiencias, tal como previamente lo hizo Martin hace más de una década.

La nueva versión Moodle 2.0 introdujo notorias mejoras en las características básicas de la primera versión, antes comentadas, corrigiendo cientos de errores que se habían ido acumulando en las sucesivas revisiones, pero es que además, introdujo otras nuevas.

Principales nuevas características de Moodle 2.0

Community hubs

- Cualquiera puede establecer un Community hub, que es un directorio de cursos para uso público o para comunidades privadas. El código es implementado como un complemento (plugin con licencia GPL) separado para Moodle.



- Los sitios pueden registrar a cualquier Community hub (en lugar de sólo moodle.org)
- Los profesores o sitios registrados pueden publicar la totalidad de sus cursos en los Community hubs, para descarga
- Los profesores en sitios registrados pueden hacer publicidad a sus cursos en Community hubs, para que las personas se unan
- Los profesores en cualquier sitio puede buscar todos los Community hubs públicos y descargar cursos como plantillas para sus propios cursos
- Usuarios en algún sitio Moodle también pueden buscar cursos en Community hubs (y comunidades de práctica) para participar en ellos. Inicialmente estamos alentando a 'comunidades de enseñanza práctica' pero cualquier conjunto de cursos puede ser listado.

Repositorios

- Moodle ahora soporta integración con repositorios externos de contenido, haciendo realmente simple llevar documentos y archivos multimedia en Moodle vía una interfaz AJAX que luce como un cuadro de diálogo estándar de Abrir en aplicaciones de escritorio.
- Los plugins iniciales en la versión 2.0 incluyen: Alfresco, Amazon S3, Box.net, el sistema de archivos propio del servidor, Flickr, Google Docs, Mahara, MERLOT, Picasa, archivos recientes, sitios remotos de Moodle, servidores WebDAV, Wikimedia, Youtube. Éstos son fáciles de desarrollar, por lo que se esperan contar con más de ellos.
- También se pueden importar archivos desde tu escritorio o especificando una dirección URL.

Portafolios

- Los módulos ahora pueden exportar sus datos a sistemas externos, particularmente útil para portafolios donde capturas instantáneas de foros, asignaciones y otras cosas en Moodle son útiles para registrar en un diario o un portafolio de evidencia



- Se soportan distintos formatos (actualmente LEAP2A, HTML, imágenes y texto, pero otros como PDF pueden ser añadidos)
- Los plugins iniciales en 2.0 incluyen: Box.net, Flickr, Google Docs, Mahara y Picasa.

Completado de cursos

- Los profesores ahora pueden especificar condiciones estándar para todos los estudiantes. Las condiciones pueden ser de completado de actividades, pero también condiciones por calificación, fecha y otros criterios.
- Los profesores pueden utilizar las condiciones como requisitos para acceder a otros cursos, lo que permite una progresión ordenada y la construcción de itinerarios.
- Los profesores y estudiantes pueden ver informes que muestran el progreso dentro del curso, o a través de una serie de cursos.

Actividades condicionales

- El acceso a actividades puede ser restringido basado en ciertos criterios como fechas, calificaciones obtenidas o la culminación de otra actividad.
- Estos pueden ser encadenados juntos para permitir revelar progresivamente el contenido del curso, si así se desea.
- Los profesores pueden especificar condiciones que definen cuándo cualquier actividad es vista como completada por un estudiante. Por ejemplo, cuando un número determinado de mensajes han sido escritos, cuando se ha alcanzado una calificación, o cuando se ha rellenado una consulta.

Cohortes o grupos masivos (Cohorts)

- También conocidos como "grupos de todo el sitio", estos son colecciones de todo el sitio de usuarios que pueden inscribirse en cursos en una sola acción, sea de forma manual o sincronizada automáticamente.

Servicios Web



- Soporte para web services basados en estándares a lo largo del código fuente de Moodle, permitiendo al administrador exponer determinadas funciones de Moodle para ser usadas por:
- Sistemas administrativos como aplicaciones de recursos humanos o sistemas de información estudiantil (SIS: Student Information System)
- Clientes móviles.
- El framework o marco de programación contiene un alto nivel de seguridad con un sistema detallado de claves o tokens y completo control sobre el rango de funciones expuestas
- Todas las funciones definidas están automáticamente disponibles vía:
 - SOAP (PHP)
 - XML-RPC
 - REST
 - AMF (Flash)

Nuevos bloques

- Bloque de comentarios - como un "shoutbox", permite comentarios que pueden ser agregados a cualquier página. Ideal para opinión de estudiantes.
- Mi bloque de archivos privados - permite fácil acceso a los archivos propios en un repositorio. de archivos privado en Moodle (con soporte de cuotas)
- Bloque de comunidad - mantiene registro de cursos externos en los que uno está interesado.
- Bloque de estado de completado de curso - reporta el estado de avance de tus cursos.

Prevención de plagio

- Moodle soporta integración con herramientas de prevención de plagio como Turnitin.

Como ya se ha dicho, Moodle es modular y ofrece varios tipos de herramientas, recursos o actividades para facilitar la docencia. Una de



esas actividades pueden ser los juegos. Y aquí entra el concepto de **serious game**.

Serious game

Constituyen un sistema de aprendizaje online, muy de moda en la actualidad. Conocidos también con el nombre de "juegos formativos", son juegos diseñados para un propósito principal distinto del de la pura diversión, orientándose fundamentalmente hacia fines sociales como la educación, la exploración científica o sanitaria, la planificación cívica, la ingeniería, etc. [2]

Las competencias requeridas por los ciudadanos para asumir un papel activo en la sociedad actual, son vistas como las habilidades del siglo XXI (21st century skills), dado que los cambios sociales los enfrentan a una renovación continua en los tipos de empleo y, consiguientemente, requieren de una educación que los habilite para trabajos que todavía no existen; y que emergerán progresivamente, propiciado por los avances tecnológicos, las nuevas formas de trabajo, etc. [3]

El desarrollo e integración de las TIC en los entornos educativos contribuye al soporte de las metodologías activas de aprendizaje basadas en la interacción entre docentes y discentes. Aunque el uso de las TIC para la enseñanza y el aprendizaje solo explotan una parte de su potencial [13], su aplicación ofrece un significativo soporte para las metodologías activas, mediante el aprendizaje basado en juegos (Game Based Learning). [3]

Los Serious Games constituyen potentes entornos para la mejora de la motivación e implicación de los participantes [14], básicamente, en el contexto específico de los juegos multi-jugadores. Romero y colaboradores (2012) [3], destacan las dinámicas de cooperación y competición que pueden tener lugar en estos contextos y, el potencial de combinación de las dinámicas de cooperación intragrupal que favorecen la interdependencia y transferencia de conocimientos, junto a las dinámicas de competición intergrupala que promueven la implicación y reto en el uso de los juegos (engagement).

Como posibles beneficios de los serious game, se pueden identificar:

- Que los cada vez más numerosos usuarios de videojuegos están sobradamente familiarizados con los procedimientos de trabajo de la plataforma.
- Que la utilización de las infraestructuras existentes para el desarrollo de los videojuegos reducen ostensiblemente el desarrollo de los propios serious game.
- Que los juegos serios no requieren de grandes soportes informáticos, exactamente igual que los videojuegos tradicionales. Su distribución se limita a enviarlos por correo o permitir su acceso mediante un sitio web dedicado.
- Y, por último, que al tiempo que los juegos serios están pensados para formar o educar a los usuarios, lo están también para entretener.



Capítulo 3.- Descripción del problema e hipótesis de trabajo

La última versión estable del módulo e-Liza desarrollado para Moodle exige una remodelación exhaustiva que cumpla con los objetivos particulares propuestos en el primer capítulo. En primer lugar, se sabe que el producto inicial no soporta las últimas versiones de la plataforma Moodle. Es labor obligada lograr esa compatibilidad. Y en segundo lugar, se dan una serie de carencias funcionales que no aprueban la herramienta para su uso en un escenario real. A continuación se exponen los tres grandes problemas de la anterior versión, los dos últimos de seguridad, así como las respectivas soluciones que pretenden darse con la nueva versión.

Problemas en e-Liza v1	Soluciones tomadas en e-Liza v2
<p>La aplicación no se consolida como una herramienta del todo ágil y cómoda para su uso. En primer lugar, como dato no tan relevante, el diseño estético no ayuda a ofrecer el carácter profesional que debería ni la sensación de aplicación robusta. En segundo lugar, cada una de las acciones llevadas a cabo por el usuario tarda algo más de lo deseado en efectuarse. Esto es porque, para cada operación, el programa se redirige a nuevas páginas (o a las mismas con distintos parámetros pasados) actualizando en cada una de ellas el contenido de la página al completo, lo que penaliza el rendimiento.</p>	<p>En e-Liza v2 la estética ha sido totalmente renovada dando una imagen divertida, elaborada y rotundamente intuitiva. Y por otro lado, las acciones de los usuarios al hacer cualquier movimiento en el programa no implican una recarga total de la página, sino que se hará uso Ajax, una de una de las nuevas tecnologías disponibles, para actualizar partes independientes de la interfaz. Así, se verá mejorada la velocidad de respuesta del módulo y con ella la agilidad de uso.</p>
<p>El fallo más importante de seguridad permite, a los alumnos que se den cuenta de ello,</p>	<p>La base del problema reside en que la comprobación de respuesta correcta o incorrecta se hace, en</p>



<p>averiguar la respuesta correcta de cada pregunta antes de ser ésta contestada.</p> <p>El proceso de contestación de una pregunta implica, en el navegador, la redirección a una nueva página pasándola algunos parámetros por método GET. Uno de estos parámetros es llamado 'bien' y puede tomar los valores 0 o 1. La lógica sencilla permite corresponder la respuesta correcta con aquella cuyo valor del parámetro 'bien' sea igual a 1. Para visualizar la página a la que será redirigida el navegador, tras dar una respuesta, así como sus parámetros pasados, basta con situar el cursor del ratón sobre dicha respuesta, sin pulsarlo, y observar la barra de estado de los principales navegadores web. En la figura situada debajo de esta tabla puede verse un ejemplo de esta violación de las reglas del juego.</p>	<p>e-Liza v1, antes de ser contestada la pregunta. Y se agrava el problema pasando los parámetros por el método GET.</p> <p>La solución más inmediata, por tanto, pasa por efectuar la comprobación de respuesta correcta o incorrecta tras la contestación de la pregunta. Además, dicha acción se llevará a cabo, de nuevo, con la tecnología Ajax por la que el navegador no se redirigirá a ninguna nueva página visible.</p>
<p>Por último, el contador que limita el tiempo dentro del cual está permitido contestar cada pregunta, puede ser reiniciado tan fácilmente como actualizar la página (pulsar F5 en los principales navegadores) en que se presenta la pregunta con el contador.</p>	<p>En la nueva versión de e-Liza, este problema será zanjado, de nuevo, con el uso de Ajax. Al no recargar la página completa cada vez que un usuario responde a una pregunta no estará permitido el método anterior para reiniciar el contador. En caso de que el usuario recargue manualmente la página global de la sección del juego o saliera de ella, quedará registrado dicho movimiento y se saldrá automáticamente de la partida en juego.</p>

Tabla 1: Problemas de e-Liza v1 y soluciones tomadas en e-Liza v2

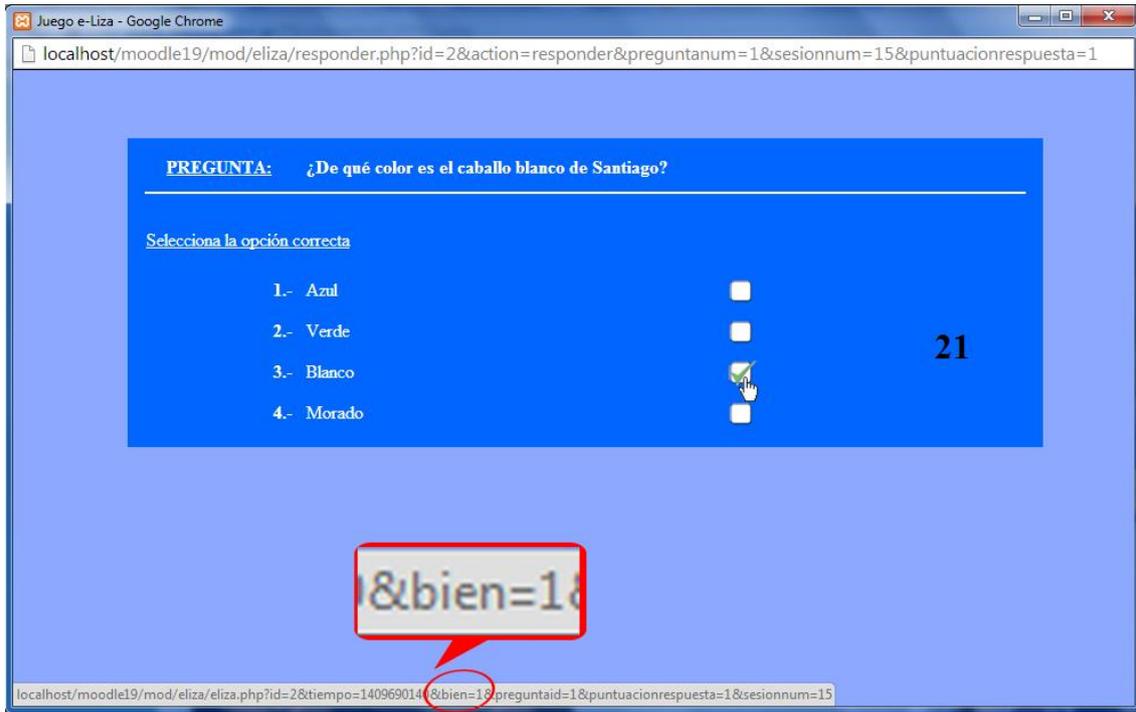


Ilustración 1: Captura del juego e-Liza v1

Es de destacar, también, que e-Liza v2 será desarrollado completamente **desde cero**, sin reutilizar fragmentos de código de los antiguos trabajos relacionados, pero sí conservando las mejores ideas encerradas y sin perder en ningún momento la esencia del módulo como juego formativo para la autoevaluación de conocimientos basado en preguntas de tipo test.

Además, ciertos aspectos técnicos serán tomados con la idea en mente de la **Internacionalización**. Por ejemplo, el nombre de las tablas y campos de la base de datos así como el nombre de las variables usadas en la programación recibirán nombres en inglés. De este modo, se consigue que futuros programadores ajenos que deseen actualizar el módulo o copiar algún fragmento de su código, puedan para ello apoyarse en el lenguaje internacional de Internet.

PARTE 2:

DESARROLLO DEL PROYECTO: e-Liza v2



Capítulo 4. Aspectos técnicos

La aplicación a desarrollar, e-Liza, no deja de ser un **módulo de actividad**, y como tal, ha de seguir unas directrices adecuadas para su completa integración con la plataforma elegida: Moodle. Es preciso tener en cuenta los posibles cambios dados con los saltos de versión de ésta. Así, aunque la forma de operar con Moodle 2.x no se aleja demasiado de la de Moodle 1.9 y sus predecesores, no hay que pasar por alto las últimas recomendaciones y los nuevos usos de funciones que bien especificados aparecen en la documentación oficial [REF]. Recordemos que la aplicación ha de ser completamente funcional para las últimas versiones de Moodle (v. 2.7) a fecha de hoy.

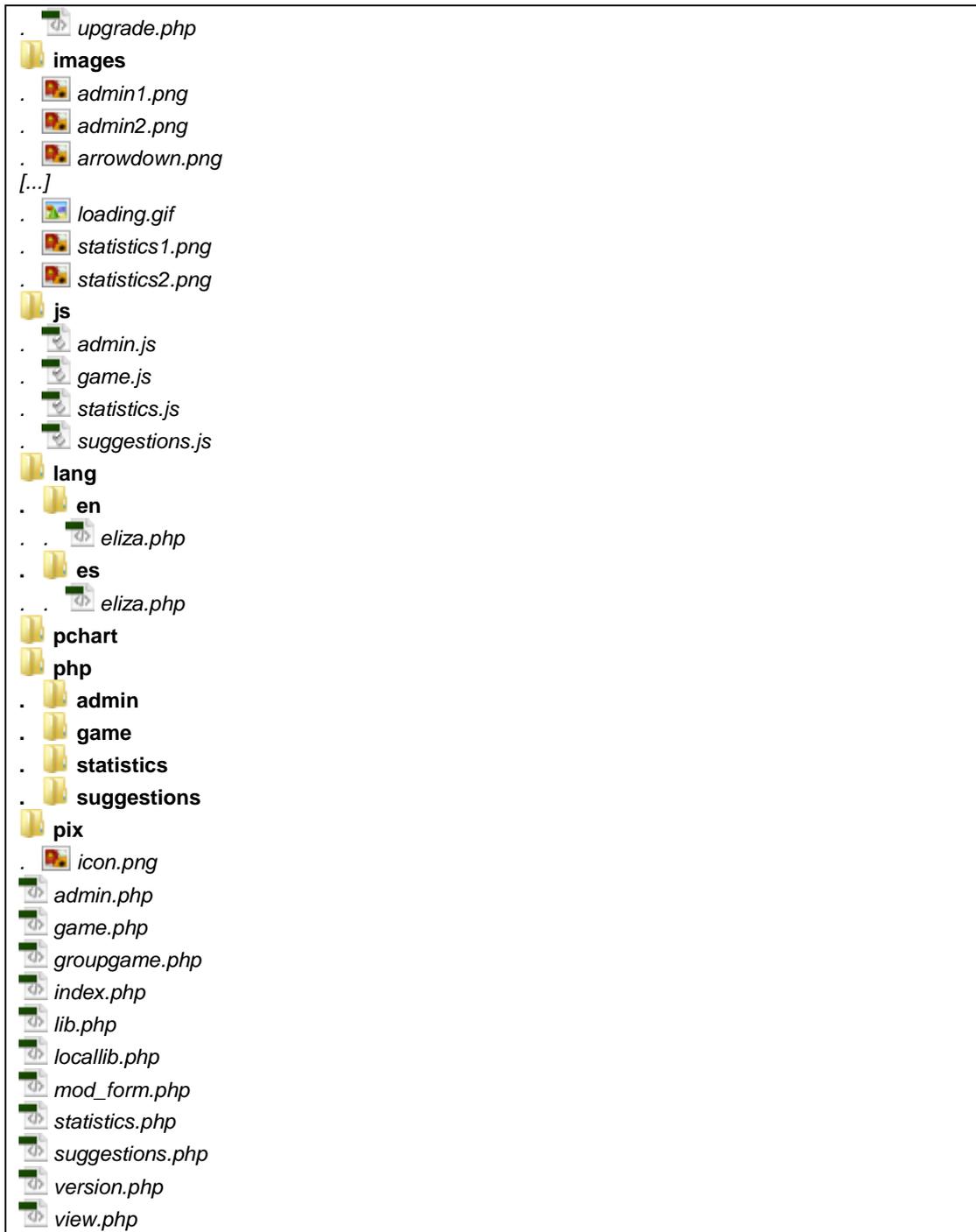
Para desarrollar un módulo de actividad en Moodle se sigue la **estructura básica de un módulo de actividad** así como un conjunto de **reglas técnicas o convenios** para el desarrollador, útiles a la hora de llevar a cabo la programación, donde se definen algunas pautas y un estilo de código.

Estructura de ficheros del módulo

Este tipo de módulos residen dentro del directorio **/mod** y están organizados en subdirectorios del mismo nombre que el del módulo en cuestión, cuyo contenido está formado por una serie de **ficheros obligatorios** junto con otros **ficheros específicos** que el desarrollador vaya a utilizar.

A continuación se presenta la estructura de ficheros del módulo e-Liza v2:





Entre ese conjunto se detallan a continuación aquellos ficheros o directorios que resultan imprescindibles en todo módulo de actividad. Se utilizan para instalar el módulo e integrarlo con el sistema Moodle y



ofrecen por separado una determinada funcionalidad que será particularizada para eliza v2 en concreto.

1. Directorio Backup

Reúne los ficheros que definen cómo se va a comportar el módulo cuando se realiza una copia de seguridad del curso o una restauración de la misma.

2. Directorio DB

2.1 Fichero access.php

Especifica las diferentes capacidades (*capabilities*) disponibles en el módulo. Esto es, define qué roles (administrador, alumno, profesor, profesor no editor, invitado, etc. [**Anexo 1**]) tienen permisos para realizar determinadas acciones, como pueden ser crear una instancia o visualizar determinado tipo de contenido.

A partir de la versión 2.3 de Moodle es recomendable añadir las capacidades:

- 'addinstance': especifica qué roles pueden crear una instancia del módulo.
- 'view': especifica el control de visibilidad de las instancias.

Para el módulo e-Liza se han creado las siguientes capacidades a mayores de las recomendadas:

- 'manage': especifica qué roles pueden administrar cada instancia del eliza editando su contenido y los parámetros del módulo. Se otorgará este permiso a los roles: 'administrador' y 'profesor con derecho de edición'.
- 'viewallstatistics': especifica qué roles pueden visualizar las estadísticas completas de la instancia del eliza, incluyendo el seguimiento de todos los usuarios participantes y no sólo la visualización del propio seguimiento del usuario identificado en el sistema. Se otorgará este permiso a los roles: 'administrador' y 'profesor con derecho de edición'.

Una vez definidas las capacidades que limitan los permisos de los diferentes roles, es posible utilizar la función

```
has_capability()
```

durante la programación para comprobar si el usuario identificado puede ejecutar una determinada acción o no. Además, la función

```
require_capability()
```

regula el control de acceso a un determinado fichero en función de la capacidad asignada al rol del usuario identificado.

2.2 Fichero install.xml

Este fichero se utiliza exclusivamente durante la instalación del módulo, donde se agregará a la base de datos la información contenida. Está escrito en lenguaje XML y define las tablas de la base de datos así como las columnas que las componen y sus propiedades.

Antes de instalar el módulo, es necesario completar adecuadamente este fichero que puede ser editado 'a mano' con cualquier editor de texto (siguiendo unas reglas específicas del estilo de código XML) o bien con ayuda del editor XMLDB que nos brinda Moodle. Éste último método es accesible desde el menú de Administración, apartado Desarrollo, y constituye una forma sencilla de generar el contenido del fichero install.xml.

El estudio de las tablas y columnas elegidas para el funcionamiento del módulo eliza puede encontrarse en este mismo capítulo en la sección 3.

2.3 Fichero upgrade.php

Se encarga de actualizar el módulo para que coincida con la última versión. Puesto que el fichero install.xml solamente se ejecuta durante la instalación, puede

darse el caso en el que al aparecer una nueva versión del módulo se necesiten nuevas tablas o nuevos campos en tablas de la base de datos que en un principio no se consideraron. Así, para cada nueva actualización, este fichero:

- Actualiza la estructura del fichero install.xml
- Agrega las nuevas tablas o columnas a la base de datos
- Actualiza el número de versión del módulo

Para el caso particular del módulo eliza en su versión 2, no se ha utilizado este fichero puesto que el desarrollo de la aplicación ha sido elaborado desde cero, pudiéndose considerar íntegramente como una versión inicial donde las tablas y sus columnas son completamente nuevas y no coinciden en absoluto con las de la versión anterior. Por ello, todo lo relativo a los cambios en la base de datos está contenido exclusivamente en el fichero install.xml. Así, para su instalación es obligatorio, en primer lugar, desinstalar por completo la primera versión del módulo y proceder a instalar la versión 2 de eliza como si de un módulo independiente se tratara.

2.4 **Directorio Lang**

Este directorio contiene la traducción de las cadenas de texto que aparecen en el módulo para los distintos posibles idiomas en que pueda estar traducido. Cada idioma está asociado a una subcarpeta cuyo nombre lo identifica (por ejemplo: directorio 'es' para idioma español, directorio 'en' para idioma inglés o directorio 'fr' para idioma francés), que contiene un fichero php del mismo nombre del módulo donde se registran todas las traducciones de las cadenas.

Cada una de estas traducciones sigue el siguiente el formato en el fichero php antes mencionado (eliza.php):

```
$string['examplechain'] = 'Cadena de ejemplo';
```

Y para utilizar dicha cadena en cualquiera de los ficheros que componen el módulo, de modo que su

traducción sea correcta y concorde al idioma seleccionado en la plataforma, se utiliza la función:

```
get_string('examplechain', 'eliza');
```

Ésta será la forma de operar a lo largo del desarrollo del módulo para asegurarse su correcta traducción e incluso permitir posibles futuras traducciones. Lo que hace de esta aplicación un sistema multilingüe.

2.5 Directorio Pix

Contiene el **icono representativo** que estará asociado al módulo en la plataforma Moodle. Será visible, por ejemplo, durante la creación de una instancia o simplemente, en la lista de actividades o recursos del curso junto con el nombre de cada instancia del módulo.

En este punto conviene tener presente que a partir de la versión 2.4 de Moodle se han dado una serie de cambios relativos a estos [15]. A tener en cuenta:

- Los nuevos iconos de actividad ahora son un poco más grandes: 24x24 píxeles, a diferencia de los antiguos: 16x16 píxeles.
- El formato de los iconos debe ser SVG (Scalable Vector Graphics) como formato recomendado o PNG en su defecto, a diferencia del antiguo formato GIF.

Además, se han especificado otras propiedades recomendadas, algo menos importantes, para diseñar los iconos de actividad:

- Borde exterior: degradado oscuro
- Borde interior: degradado claro
- Cuerpo de la imagen: degradado

Algunos ejemplos de iconos de actividad pueden verse a continuación:

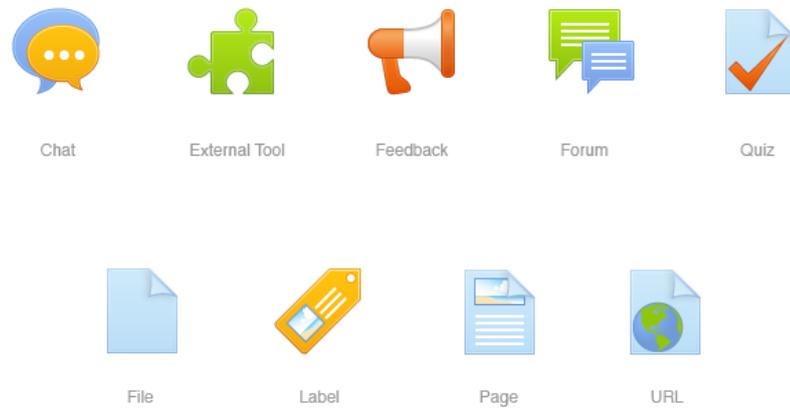


Ilustración 2: Iconos de actividad para Moodle

Por último, se dan unos ajustes SVG básicos a utilizar con el programa Adobe Illustrator que facilita el diseño.

El icono elegido para la versión 2 de eliza pretende cumplir con las nuevas especificaciones y se muestra a continuación.



Ilustración 3: Logo e-Liza v2

2.6 Fichero lib.php

Contiene las funciones básicas que necesita el módulo para su integración con Moodle así como las funciones específicas para implementar la lógica del módulo. Cada una de estas funciones comienza por 'eliza_' y es seguido por el nombre de la acción en cuestión. Como funciones obligatorias se presentan:

- eliza_add_instance(\$eliza): Añade en la base de datos una nueva instancia del módulo al crear una actividad eliza.



- `eliza_update_instance($eliza)`: Actualiza en la base de datos una instancia del módulo al editar la actividad eliza asociada.
- `eliza_delete_instance($eliza)`: Elimina de la base de datos una instancia del módulo al eliminar la actividad eliza asociada.

Si 'lib.php' se hace muy extenso, es posible utilizar el fichero 'locallib.php' para almacenar más funciones ya que ambos están conectados. El segundo llama al primero.

```
require_once("$CFG->dirroot/mod/newmodule/lib.php");
```

Sin embargo, también cabe la posibilidad de utilizar ficheros auxiliares para definir nuevas funciones o código en general necesitado por el módulo, pudiendo organizar esto a antojo del desarrollador.

2.7 Fichero mod_form.php

Define el formulario presentado al usuario para crear una instancia del módulo o para editar una existente. Así, hay que añadir al formulario todos los campos asociados a parámetros que puedan ser modificados por el creador/editor de la instancia. Para ello, ha de seguirse una sintaxis sencilla disponible en [REF5].

Una vez enviado el formulario, el módulo ejecuta la función antes mencionada `eliza_add_instance` para nuevas instancias o bien `eliza_update_instance` para actualizar instancias existentes.

Para el caso particular del módulo eliza v2 este formulario está compuesto por los siguientes campos:

- **Nombre** de la instancia
- **Introducción** descriptiva de la instancia
- **Casilla** que permite mostrar o no la anterior introducción
- **Orden de las preguntas** del juego
- Máximo **número de intentos** permitidos por usuario

- **Número de preguntas** que conformarán el juego

2.8 Fichero index.php

Esta página es utilizada por Moodle para enumerar en forma de tabla todas las instancias del módulo eliza relativas a un mismo curso.

2.9 Fichero view.php

Constituye la página principal de una actividad eliza. Es decir, representa el menú principal de una instancia del módulo. Como es lógico, tiene diferente aspecto para los alumnos que para los profesores con derecho de edición del curso:



Ilustración 4: Menú principal para el rol profesor



Ilustración 5: Menú principal para el rol alumno

2.10 Fichero versión.php

Almacena la versión del módulo y otros atributos, como la versión de Moodle mínima requerida para su instalación. Será suficiente con éstos para el módulo eliza v2, donde se especifica una versión mínima de Moodle requerida de 2.7.0 y una versión inicial del módulo en sí, que data de Agosto de 2014:

```
$plugin->requires = 2014051200; // Moodle 2.7.0
$plugin->version = 2014082000; // 20/8/2014
```

Y una vez definidos los ficheros y directorios comunes a todo módulo de actividad, particularizados para eliza v2, se exponen a continuación los **ficheros** o **directorios específicos** del módulo que completan la estructura de ficheros. Cabe mencionar que se ha seguido una organización simple y clara de la misma con el objetivo de permitir a posibles desarrolladores continuar con el trabajo en un futuro, si se desea.

2.11 Directorio css

Contiene el fichero style.css donde se almacena toda la información relativa al estilo de la aplicación. Cualquier cambio de apariencia del módulo puede efectuarse aquí.

2.12 Directorio Images

Reúne todas las imágenes utilizadas por el módulo. Todas ellas, salvo una, están en formato PNG (Portable Network Graphics) aprovechando en muchas de ellas la característica de transparencia que nos brinda dicho formato así como su buena compresión. Además, mantienen un modo entrelazado que permite mostrar la imagen en el navegador de forma gradual, de modo que si la carga es lenta, al menos la imagen aparece completa de inicio mejorando progresivamente su calidad, dando una idea al usuario de lo que se va a mostrar finalmente.

La única imagen que mantiene un formato GIF aprovecha su característica de animación para

representar un icono que hace referencia a la carga de alguna parte del contenido de la página.

2.13 Directorio js

Contiene cinco ficheros (uno para cada sección del módulo) escritos en lenguaje javascript donde se almacenan las funciones de este lenguaje utilizadas. La mayoría de estas funciones constituyen llamadas Ajax, que permiten recargar contenido de una determinada parte de la página sin recargar la página al completo.

2.14 Directorio php

Contiene cinco subdirectorios (uno para cada sección del módulo) donde se almacenan ficheros escritos en lenguaje PHP, que dan respuesta a las llamadas Ajax antes mencionadas o a determinadas operaciones del módulo.

2.15 Directorio pchart

Almacena los ficheros de la librería del mismo nombre [16]. Esta librería se integra con PHP y permite mostrar gráficas de distintos tipos a partir de unos datos dados. Será utilizada para representar gráficamente algunas de las estadísticas del juego en la sección de estadísticas precisamente.

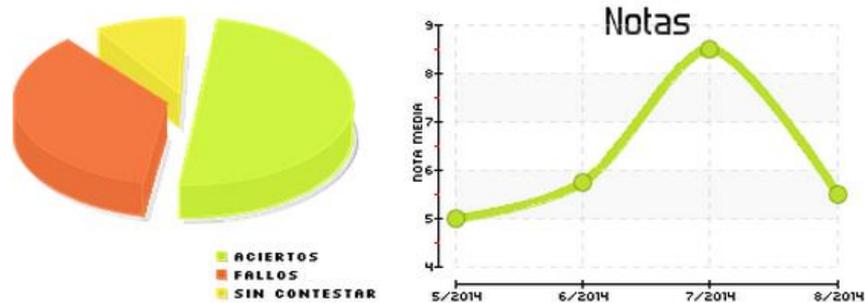


Ilustración 6: Gráficas generadas con pChart

Por último, los siguientes ficheros están asociados a la representación de cada una de las secciones de la aplicación,



cuyas funcionalidades serán detalladas en la segunda sección de este capítulo.

2.16 Fichero `admin.php`

Se corresponde con la sección de administración de la aplicación. Evidentemente solamente es accesible por un profesor con derecho de edición del curso.

2.17 Fichero `game.php`

Se corresponde con la sección del juego en sí. Es accesible por cualquier usuario del curso y será utilizado por los alumnos para resolver el test, a modo de juego, propuesto por el profesor.

2.18 Fichero `statistics.php`

Se corresponde con la sección de estadísticas de la aplicación. Si el usuario identificado es un profesor, podrá visualizar todas las estadísticas relativas al eliza incluyendo las estadísticas de todos los usuarios y de todas las preguntas. Si el usuario identificado es un alumno, podrá visualizar únicamente sus propias estadísticas.

2.19 Fichero `suggestions.php`

Se corresponde con la sección de propuestas de preguntas de la aplicación, donde cualquier alumno será capaz de gestionar sus propias propuestas de preguntas para el juego.

2.20 Fichero `groupgame.php`

Se corresponde con la sección que permite el juego entre alumnos, a modo de competición. Es accesible por cualquier usuario del curso y será utilizado por los alumnos para retar a otros a superar su nota en una única partida.

Para facilitar el trabajo al desarrollador, es posible partir de una **plantilla básica** accesible desde [17] y adaptada a las versiones 2.x de Moodle. Ésta contiene perfectamente organizados en sus rutas adecuadas, los ficheros imprescindibles, antes explicados, que conforman el módulo como una aplicación de esta plataforma.



Pautas de programación

Reglas generales

Resulta imprescindible contemplar una serie de reglas lógicas de carácter general, antes de empezar a escribir cualquier fragmento de código, para asegurar la posterior integración del nuevo módulo con cualquier sistema Moodle. Eliza v2 ha sido desarrollado cumpliendo, en todo momento, las principales pautas estandarizadas [18]:

1. Todos los archivos de código deberían utilizar la extensión .php.
2. Todas las plantillas deberían utilizar la extensión .html.
3. Todos los avisos de copyright deben ser mantenidos.
4. Todos los archivos deben incluir el archivo principal config.php.
5. Cualquier otro include/require debería utilizar una ruta absoluta que comience por \$CFG->dirroot o \$CFG->libdir, nunca relativos.
6. Cada archivo debería comprobar que el usuario está autenticado correctamente.
7. Todos los accesos a la base de datos deberían utilizar las funciones definidas en lib/datalib.php cuando sea posible - esto permite la compatibilidad con un gran número de bases de datos-. Toda la documentación relativa a XMLDB, la capa de abstracción de la base de datos que utiliza Moodle, puede encontrarse en [REF]. Así, para la gestión de los datos desde el módulo, pueden diferenciarse dos tipos de funciones basadas en SQL, que serán de uso obligatorio al realizar las consultas:
 - DDL (Data Manipulation Language): para gestionar las estructuras de las tablas. No serán utilizadas por eliza v2.
 - DML (Data Manipulation Language): para gestionar el contenido de las tablas.
8. No crear o utilizar variables globales distintas de las estándar \$CFG, \$SESSION, \$THEME, \$SITE, \$COURSE y \$USER.



9. Todas las variables deberían ser inicializadas o, al menos, comprobada su existencia utilizando `isset()` o `empty()` antes de ser usadas.
10. Todas las cadenas deberían ser traducibles.
11. Todos los errores deberían ser visualizados utilizando la función `print_error()` para maximizar la traducción y ayudar a los usuarios (automáticamente se enlaza con moodle docs).
12. Todos los ficheros de ayuda deben ser traducibles.
13. Muy importante: todos los textos dentro de moodle, especialmente aquellos que han sido introducidos por los usuarios, deben ser mostrados utilizando la función `format_text()`. Esto asegura que el texto es filtrado y limpiado correctamente.
14. Las acciones de los usuarios deberían ser registradas utilizando la función `add_to_log()`.
15. Los enlaces html, deben ser siempre relativos a la raíz del sitio moodle.

Estilo de código

Mantener un estilo de código consistente es importante en cualquier proyecto de desarrollo y, en particular, en aquellos donde muchos desarrolladores están involucrados. Un estilo estándar ayuda a asegurar que el código es más fácil de leer y entender, lo que permite que distintos programadores puedan trabajar en arreglos o futuras nuevas versiones del módulo. Así, eliza v2 ha sido elaborado teniendo en mente las pautas más importantes [19]:

1. Utilizar siempre las etiquetas completas de php: `<?php ?>` y no reducidas: `<? ?>`
2. Utilizar una sangría de 4 espacios sin caracteres de tabulación y evitar la sangría en el primer nivel del código.
3. Escribir una longitud máxima por línea de 180 caracteres, siendo de 138 el máximo recomendable.
4. Las terminaciones de línea deben seguir el formato de texto estándar de UNIX donde cada línea debe terminar con un



salto de línea (linefeed, LF), representado como el número ordinal 10 o hexadecimal 0x0A, y no utilizar retornos de carro (CR) propios de los viejos ordenadores Macintosh ni la combinación de retornos de carro y saltos de línea (CRLF) propios de los ordenadores Windows.

5. Las líneas de código no deben tener espacios finales.
6. Los nombres de archivos:
 - a. Deben ser íntegramente palabras inglesas
 - b. Deben ser tan cortos como sea posible
 - c. Deben contener únicamente letras minúsculas
 - d. Deben terminar en extensiones php, html, css, js o xml.
7. Los nombres de las clases deben estar compuestos por palabras en minúsculas y en inglés, separadas por guiones bajos.
8. Los nombres de las funciones deben estar compuestos por palabras en minúsculas y en inglés, separadas por guiones bajos y deben estar precedidos por un prefijo correspondiente al nombre del módulo. Se debe evitar también que no haya un espacio entre el nombre, la función y el paréntesis. Por ejemplo: `eliza_add_instance()`.
9. Los argumentos de las funciones deben estar separados por un único espacio posterior, después de la coma que los delimita.
10. Los nombres de las variables deben estar compuestos por palabras en minúsculas, en inglés, fáciles de leer y sin separaciones, si hubiera varias.
11. Los nombres de las constantes deben estar compuestos por palabras en mayúsculas y en inglés, separadas por guiones bajos y deben estar precedidos por un prefijo correspondiente al nombre del módulo.
12. Usar letras minúsculas para escribir valores booleanos: `true`, `false` y `null`.
13. Los bloques de código siempre deben estar encerrados por llaves (incluso si sólo constan de una línea).
14. Las cadenas tienen que ser definidas utilizando comillas simples, siempre que sea posible, para obtener un mejor rendimiento.



15. Los comentarios deben ser añadidos de forma que resulten prácticos, para explicar el flujo del código y el propósito de las funciones y variables. Los comentarios de una sola línea deben usar dos barras verticales seguidas de un espacio.

Pautas para la estructura de la base de datos

Sabiendo que Moodle tiene una poderosa capa de abstracción llamada XMLDB, conviene utilizar adecuadamente las herramientas sugeridas, tanto para definir y modificar tablas como para realizar consultas sobre ellas con el objetivo de conseguir que un mismo código que interactúe con la base de datos funcione correctamente con independencia del tipo de sistema gestor de ésta (MySQL/MariaDB, PostgreSQL, MS SQL Server u Oracle). Para el desarrollo de eliza v2 se ha tenido en cuenta correctamente la documentación de XMLDB [REF] así como una serie de pautas que estandarizan la estructura global de la base de datos de la plataforma y se exponen a continuación [20]:

1. Cada tabla debe tener un campo autonumérico id (INT10) como clave primaria.
2. La tabla principal que contiene instancias de cada módulo debe tener el mismo nombre que el módulo (eliza) y contener, por lo menos, los siguientes campos:
 - id: descrito arriba
 - course: el identificador del curso al que la instancia pertenece
 - name: el nombre completo de la instancia
3. El resto de las tablas asociadas con el módulo deberían ser llamadas eliza_name, donde name se corresponde con el nombre de los elementos que almacene dicha tabla en plural. Por ejemplo la tabla que contiene todas las preguntas del eliza se llama eliza_questions.
4. Los nombres de las tablas y de los campos tienen que evitar el uso de palabras reservadas por las Bases de Datos.



5. Los nombres de los campos (columnas) deberían ser sencillos y cortos, siguiendo las mismas reglas que los nombres de las variables.
6. Cuando sea posible, las columnas que contengan una referencia al campo id de otra tabla (por ejemplo, eliza) debería ser llamado elizaid (esta norma es nueva y puede no ser seguida por algunas tablas antiguas).
7. Los campos booleanos serán implementados como enteros cortos (por ejemplo, INT4) con los valores 0 o 1, para permitir la futura expansión de los valores, si fuera necesario.
8. La mayoría de las tablas tienen que tener un campo timemodified (INT10), que será actualizado con la fecha actual (timestamp de UNIX) obtenida con la función time() de PHP.
9. Conviene definir siempre un valor por defecto para cada campo.
10. Cada tabla debe comenzar con el prefijo de la base de datos (\$CFG->prefix). En muchos casos esto es gestionado automáticamente.
11. Seguir las reglas siguientes sobre el uso del comando AS:
 - No utilizar el comando AS para alias de tablas.
 - Utilizar el comando AS para alias de campos (columnas).
12. Nunca crear UNIQUE KEYs (restricciones) para nada. En su lugar, utilizar UNIQUE INDEXes.
13. Cuando se haga referencia a una variable entera en consultas SQL, no ha de entrecomillarse el valor.

Normas de seguridad

En este punto, se especifican una serie de hábitos encarecidamente recomendables que se han seguido y persiguen evitar las posibles vulnerabilidades de la plataforma Moodle causadas por el módulo eliza v2. Esto es importante puesto que se pone en peligro la integridad y confidencialidad de los datos asociados a los usuarios de la plataforma [21].



1. Autenticar al usuario en todas las páginas con la función `require_login` o bien `require_course_login`.
2. Especificar correctamente el curso al comprobar la protección de las áreas del curso.
3. Comprobar los permisos del usuario identificado ante cualquier acción que lo necesite con las funciones `has_capability` y `require_capability`.
4. No basarse en `'register_globals'`. Cada variable debe ser correctamente inicializada en cada fichero de código. Debe ser obvia la procedencia de cada variable.
5. Inicializar todos los arrays y objetos aunque estén vacíos. `$a = array()` o `$obj = new stdClass();`.
6. No utilizar la función `optional_variable()`. En su lugar, utilice la función `optional_param()` para recuperar parámetros enviados por GET o POST. Seleccione la opción `PARAM_XXXX` apropiada al tipo de parámetro que espera. Para comprobar y definir un valor opcional para una variable, utilice la función `set_default()`.
7. No utilizar la función `require_variable()`. En su lugar, utilice la función `required_param()` para recuperar parámetros enviados por GET o POST y además sea necesario su envío. Seleccione la opción `PARAM_XXXX` apropiada al tipo de parámetro que espera.
8. No utilizar `$_GET`, `$_POST` o `$_REQUEST`. En su lugar, utilice las funciones `required_param()` o `optional_param()` apropiadas.
9. No comprobar las acciones con código como: `if (isset($_GET['algo']))`. Utilice, por ejemplo, `$algo = optional_param(algo, -1, PARAM_INT)` y entonces compruebe que está dentro de los valores esperados, por ejemplo, `if ($something>=0) {...`
10. Cuando sea posible, agrupe todas sus llamadas a `required_param()`, `optional_param()` y el resto de inicialización de variables en el principio de cada fichero (o función) para que sea fácilmente localizable.
11. Cualquier información leída desde la base de datos debe tener la función `addslashes()` aplicada antes de volver a enviar



la información a la base de datos. Un objeto completo puede ser procesado con la función `addslashes_object()`.

12. No utilice información obtenida de `$_SERVER` si puede evitarlo. Presenta algunos problemas de portabilidad.
13. Si escribe código SQL, asegúrese completamente de que es correcto. En particular, compruebe la falta de comillas en las variables utilizadas. Es un punto de entrada para ataques de tipo 'SQL injection'.
14. Comprobar toda la información (especialmente la que es enviada a la base de datos) que es utilizada en cada fichero. Nunca confíe en que otro código estará haciendo ese trabajo.

Diseño de la base de datos

Uno de los fundamentos de la aplicación consiste en almacenar organizadamente los datos relativos a la misma, para después operar con ellos con cuatro posibles tareas:

- Añadir nuevos datos
- Borrar datos
- Actualizar datos
- Recuperar datos

Moodle trabaja con **sistemas relacionales** de bases de datos. Por tanto, los datos se agrupan en tablas bidimensionales interrelacionadas. Y el módulo eliza v2 ha de encargarse de almacenar adecuadamente toda la información asociada a las preguntas del juego, propuestas de los alumnos, estadísticas individuales, etc.

A la hora de diseñar la base de datos, ha sido necesario **normalizar**, proceso que sigue una serie de pasos o normas de forma que, tras aplicar todas ellas, se obtiene la descomposición de los datos en diferentes tablas cuya estructura se considera óptima para su implementación, gestión y explotación. La normalización tiene tres etapas y todas han sido completadas con éxito para el diseño de la base de datos de eliza v2 [22]:



- **Primera forma normal** (1FN): Evitar que en los atributos aparezcan varios valores; éstos tienen que ser elementales y únicos.
- **Segunda forma normal** (2FN): Aislar en una tabla todos los datos que sólo dependen de una parte de la clave, junto con dicha parte de la clave.
- **Tercera forma normal** (3FN): Deshacerse de cualquier cosa de cada tabla que no dependa únicamente de la clave primaria.

Así, un buen diseño de la base de datos resulta un trabajo imprescindible que tiene por principales objetivos:

- Evitar inconsistencias y problemas de integridad
- Eliminar o minimizar la redundancia en la información almacenada
- Sostener un control centralizado de la información, permitiendo un máximo rendimiento a la plataforma cuando accede a los datos

Teniendo todo esto en cuenta, la estructura de tablas y sus campos pertenecientes al módulo desarrollado así como sus relaciones entre sí se presentan a continuación.

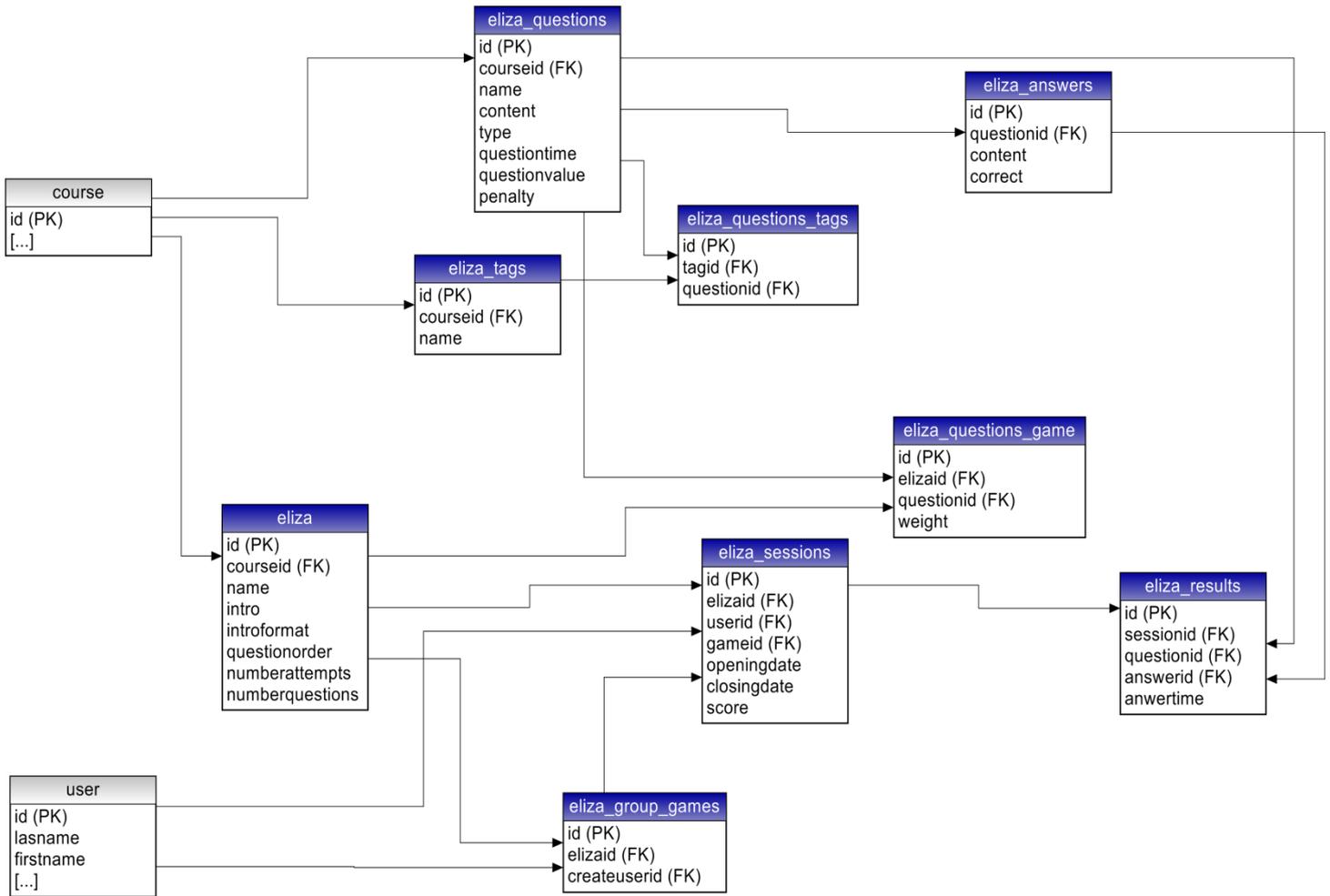


Ilustración 7: Esquema de la base de datos

Pese a que las tablas *Course* y *User* no forman parte del módulo en sí, se han incluido en el esquema puesto que mantienen alguna relación con tablas que sí forman parte del módulo, lo que permite acceder a una información algo más detallada del curso y de los usuarios registrados en la plataforma desde alguna parte de eliza v2.

A continuación se detalla la estructura de cada una de las tablas:

Tabla eliza

Se trata de la tabla principal del módulo. Almacena los datos relativos a cada una de las instancias del módulo, esto es, los parámetros principales asociados a cada juego creado como actividad de un curso en la plataforma. El número de registros que contendrá coincidirá con el número de juegos creados en toda la plataforma educativa.

eliza				
Campo	Tipo	Restricción	Predet.	Comentario
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de la instancia
course	bigint(10) [Foreign Key]	NOT NULL	-	Identificador del curso al que pertenece la instancia
name	varchar(255)	NOT NULL	-	Nombre de la instancia
intro	longtext	NULL	NULL	Descripción de la instancia
introformat	smallint(4)	NOT NULL	0	Activar formato a la descripción
timecreated	bigint(10)	NOT NULL	-	Fecha de creación de la instancia
timemodified	bigint(10)	NOT NULL	0	Fecha de modificación de la instancia
questionorder	varchar(16)	NOT NULL	-	Orden de las preguntas
numberattempts	bigint(10)	NOT NULL	-	Máximo número de intentos permitidos por usuario para jugar
numberquestions	varchar(20)	NOT NULL	-	Número de preguntas de que estará

compuesto el juego

Tabla 2: eliza

El campo **questionorder** debe almacenar la cadena 'RANDOM', si se desea que las preguntas aparezcan en el juego con un orden aleatorio o bien la cadena 'SEQUENTIAL', si se desea que las preguntas aparezcan en el juego siguiendo un orden marcado por el profesor. Si las preguntas aparecen en orden aleatorio, está asegurado que ninguna de ellas se repetirá si no ha acabado la ronda completa de todas las seleccionadas por el profesor para la instancia del eliza.

El campo **numberattempts** limita el número de intentos permitidos por usuario para resolver una misma instancia de eliza o juego. Si este campo es 0, el número de intentos será ilimitado.

El campo **numberquestions** define el número de preguntas que compondrán el juego. Si el profesor ha seleccionado un mayor número de preguntas para la instancia del eliza que el número que almacena este campo, entonces se verá limitado apareciendo durante el juego únicamente este número de preguntas indicado. Este campo también puede tomar el valor 'ALL' para ordenar que se muestren en el juego todas las preguntas que el profesor haya seleccionado para la instancia del eliza o el valor 'UNLIMITED' para ordenar que se muestren ilimitadas preguntas en el juego, repitiéndose las mismas, si fuera necesario, pero ordenadas de forma que se repitan lo menos posible: En primer lugar, una ronda de todas las seleccionadas, en segundo lugar, otra ronda, y así sucesivamente, sin repetir ninguna pregunta por ronda.

Se recuerda que solamente aparecerán en el juego las preguntas que el profesor haya seleccionado del banco de preguntas para la instancia del eliza.

Tabla eliza_questions

Almacena toda la información relativa a cada una de las preguntas existentes en el módulo creadas por los profesores con derechos de edición del curso, que serán los usuarios con permisos para ello.

Al agrupar las preguntas en un banco de preguntas global y unívocamente relacionado con cada curso, será necesario almacenar para cada pregunta el identificador del curso en cuyo banco de preguntas debe estar incluida.

eliza_questions				
Campo	Tipo	Restricción	Predet.	Comentario
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de la pregunta
createuser	bigint(10) [Foreign Key]	NULL	NULL	Identificador del usuario creador
courseid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador del curso
name	varchar(255)	NOT NULL	-	Nombre de la pregunta
content	longtext	NOT NULL	-	Contenido de la pregunta. Pregunta en sí misma
type	varchar(15)	NOT NULL	GENERIC	Tipo de pregunta
questiontime	bigint(10)	NULL	NULL	Tiempo disponible para dar respuesta
questionvalue	decimal(10,2)	NULL	1.00	Valor de la pregunta. Puntuación.
penalty	int(7)	NULL	0	Porcentaje de questionvalue a restar en caso de fallo

Tabla 3: eliza_questions

Tabla eliza_answers

Esta tabla reúne toda la información relativa a las respuestas asociadas a las preguntas. Cada respuesta puede pertenecer únicamente a una pregunta, por lo que esta misma tabla integra un campo que almacena el identificador de la pregunta a la que pertenece la respuesta. Sin embargo, una pregunta puede tener indefinidas respuestas asociadas.



Por esta razón, se ha creado una nueva tabla para almacenar expresamente las respuestas en lugar de haberlo hecho en la propia tabla de preguntas (eliza_questions).

eliza_answers						
Campo	Tipo	Restricción	Predet.	Comentario		
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de respuesta	de	la
questionid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de pregunta asociada	de	la
Content	longtext	NOT NULL	-	Contenido de respuesta	de	la
correct	tinyint(1)	NULL	NULL	Booleano para comprobación de respuesta	de	la

Tabla 4: eliza_answers

Tabla eliza_questions_games

Dado que una pregunta puede ser reutilizada para elaborar distintas instancias de eliza (juegos), no existe una relación unívoca entre un juego y una pregunta. Una pregunta puede estar integrada en varios juegos diferentes y un juego puede integrar varias preguntas distintas. Por esto, se necesita una relación **muchos a muchos** que esta tabla se encargará de almacenar.

Además, se precisa de algún método que permita controlar el orden de aparición de las preguntas, siempre y cuando el profesor no desee un orden aleatorio, de modo que se presenten de un modo secuencial y completamente a elección. El campo **weight** (peso) de esta tabla cumple esta función. De modo que para una misma instancia de eliza (juego), las preguntas con menor peso aparecerán antes que las de mayor peso (siempre y cuando, recordemos, el parámetro questionorder de la tabla eliza almacene la cadena 'SEQUENTIAL' y no 'RANDOM', con la que las preguntas aparecerían en orden aleatorio. De lo contrario, este campo no tendrá ningún efecto).

Tabla eliza_tags

Esta tabla almacena simplemente el nombre de las posibles etiquetas que pueden asociarse a las distintas preguntas. Para cada curso diferente hay disponibles un conjunto de etiquetas totalmente administrables por el profesor. El campo **courseid** registra precisamente el identificador del curso al que pertenece la etiqueta.

eliza_tags				
Campo	Tipo	Restricción	Predet.	Comentario
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de la etiqueta
courseid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador del curso
name	varchar(255)	NOT NULL	-	Nombre de la etiqueta

Tabla 5: eliza_tags

Tabla eliza_tags_questions

Puesto que una pregunta puede ser etiquetada con indefinidas etiquetas y una etiqueta puede ser utilizada para etiquetar varias preguntas diferentes, se contempla una **relación muchos a muchos** que hace necesaria la creación de esta tabla. Aquí se almacenan las relaciones existentes entre las preguntas del curso y sus etiquetas asociadas.

eliza_tags_questions				
Campo	Tipo	Restricción	Predet.	Comentario
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de la relación
tagid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de la etiqueta



Field	Type	Constraint	Default	Description
questionid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de la pregunta

Tabla 6: eliza_tags_questions

Tabla eliza_suggestions

Almacena toda la información relativa a cada una de las preguntas existentes en el módulo, creadas por cualquier alumno con ánimo de proponer al profesor dicha pregunta para ser agregada al banco de preguntas del curso.

El campo **createuser** registra el identificador del usuario que ha creado la propuesta de pregunta. Y el campo **courseid** es necesario para identificar el curso al que esa propuesta de pregunta pertenece.

Cuando una propuesta de pregunta es aprobada por el profesor, algunos datos del registro de esta tabla serán copiados automáticamente a la tabla eliza_questions, donde finalmente quedará integrada la nueva pregunta en el banco de preguntas del curso.

El campo **state** indica el estado en que se encuentra la propuesta ofrecida por el alumno. Así, cuando el estado es 'PENDING' quiere decir que aún está pendiente de aprobación, cuando el estado es 'APPROVED' quiere decir que la propuesta ha sido aprobada por el profesor y agregada al banco de preguntas del curso, y cuando el estado es 'REJECTED' quiere decir que el profesor ha rechazado la propuesta de pregunta.

eliza_suggestions				
Campo	Tipo	Restricción	Predet.	Comentario
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de la propuesta
createuser	bigint(10) [Foreign Key]	NULL	NULL	Identificador del usuario creador de la propuesta
courseid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador del curso



name	varchar(255)	NOT NULL	-	Nombre de la pregunta
content	longtext	NOT NULL	-	Contenido de la pregunta. Pregunta en sí misma
type	varchar(15)	NOT NULL	GENERIC	Tipo de pregunta
questiontime	bigint(10)	NULL	NULL	Tiempo disponible para dar respuesta
questionvalue	decimal(10,3)	NULL	1.000	Valor de la pregunta. Puntuación.
penalty	int(7)	NULL	0	Porcentaje de questionvalue a restar en caso de fallo
suggestiondate	int(11)	NULL	NULL	Fecha de creación de la propuesta
state	varchar(32)	NULL	NULL	Estado de la propuesta

Tabla 7: eliza_suggestions

Tabla eliza_suggestions_answers

Esta tabla reúne toda la información relativa a las respuestas asociadas a las propuestas de preguntas. Cada respuesta puede pertenecer únicamente a una propuesta, por lo que esta misma tabla integra un campo que almacena el identificador de la propuesta a la que pertenece la respuesta. Sin embargo una propuesta puede tener indefinidas respuestas asociadas. Por esta razón, se ha creado una nueva tabla para almacenar expresamente las respuestas en lugar de haberlo hecho en la propia tabla de propuestas de preguntas (eliza_suggestions).

eliza_suggestions_answers				
Campo	Tipo	Restricción	Predet.	Comentario

id	int(11) [Primary Key]	NOT NULL	-	Identificador de la respuesta
suggestionid	int(11) [Foreign Key]	NOT NULL	-	Identificador de la pregunta propuesta
content	text	NULL	NULL	Contenido de la respuesta
correct	tinyint(1)	NULL	NULL	Booleano para la comprobación de la respuesta

Tabla 8: eliza_suggestions_answers

Tabla eliza_sessions

Uno de los puntos fuertes de eliza v2 es el control absoluto de las estadísticas. Para esto, es necesario registrar cada movimiento del usuario. Llamaremos **sesión** a cada una de las veces que un usuario comienza un nuevo juego correspondiente a una instancia de eliza. Así, esta tabla almacena todas las sesiones de todas las instancias del módulo.

Cuando un usuario comienza un juego quedan registradas la fecha y hora en la que lo hace en el campo **openingdate**. Del mismo modo, cuando el usuario finaliza el juego quedan registradas la fecha y hora de cierre de sesión en el campo **closingdate**. Si el usuario no llega a acabar la sesión, éste último campo quedará vacío y se verá reflejado en las estadísticas que dicha sesión no ha sido cerrada, lo que puede ser de utilidad para el profesor.

El campo **gameid**, por su parte, sólo tiene sentido cuando la sesión proviene de un juego en grupo almacenando aquí el identificador de la partida en grupo asociada.

Y el campo **score** registra la nota sobre 10 obtenida por el alumno en la sesión. En principio, este campo es totalmente prescindible puesto que es posible calcular dicha nota a partir de los datos de las preguntas acertadas, falladas o no contestadas en la sesión (almacenadas en la siguiente tabla). Sin embargo, se ha decidido registrar este valor puesto

que si en algún momento se elimina alguna de esas preguntas del banco de preguntas, y con ello su valor, se perdería la contabilidad de la nota real que el alumno obtuvo al llevar a cabo la sesión.

eliza_sessions				
Campo	Tipo	Restricción	Predet.	Comentario
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de la sesión
userid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador del usuario
elizaid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de la instancia eliza
gameid	bigint(10) [Foreign Key]	NULL	NULL	Identificador de la partida en grupo
ipadress	varchar(255)	NULL	NULL	Dirección IP del ordenador
openingdate	varchar(255)	NULL	NULL	Fecha de apertura de la sesión
closingdate	varchar(255)	NULL	NULL	Fecha de cierre de la sesión
score	float	NULL	NULL	Puntuación obtenida en la sesión

Tabla 9: eliza_sessions

Tabla eliza_results

Esta tabla reúne todas las relaciones de preguntas aparecidas y sus respuestas dadas por el alumno durante de las sesiones de la tabla anterior. Es decir, cada vez que un alumno responde a una pregunta durante alguna de sus partidas, automáticamente quedarán registrados en esta tabla el identificador de la pregunta y el identificador de la respuesta dada por el usuario. Asimismo quedará guardado en el campo **answertime** el tiempo en segundos que el alumno ha tardado en contestar.

Puesto que cada uno de estos registros está asociado a una sesión, el campo **sessionid** se encarga de llevar a cabo este vínculo.

eliza_results				
Campo	Tipo	Restricción	Predeterminado	Comentario
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de la relación
questionid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de la pregunta
sessionid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de la sesión
answerid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de la respuesta
answertime	decimal(10,2)	NULL	NULL	Tiempo tardado en dar respuesta

Tabla 10: eliza_results

Tabla eliza_group_games

Almacena la información relativa a las partidas en grupo que pueden generarse por cualquier alumno. El campo **elizaid** asocia cada partida en grupo con la instancia del módulo eliza a la que pertenece.

eliza_group_games				
Campo	Tipo	Restricción	Predet.	Comentario
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador de la partida en grupo
elizaid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de la instancia eliza
creatoruserid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador del usuario creador de la

	Key]			partida	
name	varchar(255)	NOT NULL	-	Nombre de la	partida
createdate	bigint(10)	NOT NULL	-	Fecha de creación	de la partida

Tabla 11: eliza_group_games

Tabla eliza_group_games_users

Puesto que cada juego en grupo puede ser integrado por un número indefinido de participantes y cada participante puede ingresar en varios grupos diferentes, se hace necesaria la creación de esta tabla para almacenar la relación entre los distintos juegos en grupo y sus participantes inscritos.

eliza_group_games_users					
Campo	Tipo	Restricción	Predet.	Comentario	
id	bigint(10) [Primary Key]	NOT NULL	-	Identificador	de la relación
gameid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador de la partida	
userid	bigint(10) [Foreign Key]	NOT NULL	-	Identificador del usuario	

Tabla 12: eliza_group_games_users

Capítulo 5. Aspectos funcionales

En este punto de la memoria se procede a detallar las características que ofrece eliza v2 para cumplir su cometido. No deja de ser un juego formativo donde se examina al alumno con preguntas de tipo test totalmente personalizables. Cada una de las **funcionalidades de la aplicación** puede presentarse separando las distintas **secciones** de que consta el programa, que coinciden, dicho sea de paso, con los apartados del menú principal. Así, organizamos los aspectos funcionales del mismo modo en que está organizado el menú del programa.



Ilustración 8: Secciones del programa

Jugar

Esta funcionalidad está orientada al uso de la aplicación por parte del alumno. En primera instancia se muestran en una tabla los **parámetros del juego**, que previamente han sido definidos por el profesor. A mayores del nombre del usuario que va a comenzar el juego, se muestran:

- El **orden de las preguntas**, que puede ser aleatorio o secuencial, indicando así que la aparición de las preguntas del juego seguirá tal orden.
- El **número de intentos** que el jugador habrá empleado para jugar a la instancia del e-Liza así como el número máximo de intentos permitidos para ello. Puede indicar que se permiten ilimitados intentos o un número entero de ellos.
- El **número de preguntas** que compondrán el juego. De nuevo puede indicar que el número de preguntas es ilimitado, caso en el que el jugador podrá abandonar el juego tras contestar



cualquiera de ellas, o puede indicar, en su defecto, un número entero de preguntas que conformarán el juego.

En el caso de que las preguntas sean ilimitadas, aparecería en primer lugar una primera ronda de todas las preguntas que componen la instancia del e-Liza sin repetir ninguna de ellas y en el orden indicado previamente. A continuación, se mostraría una segunda ronda de las mismas preguntas de modo que si el orden de las preguntas fuera aleatorio, se reordenarían aleatoriamente las preguntas y así para cada nueva ronda mostrada. Si el orden de las preguntas fuera secuencial, todas las rondas de preguntas seguirían el mismo orden de las preguntas que previamente ha sido fijado por el profesor.

Esta forma de operar asegura que se repitan el mínimo número posible de preguntas durante el juego.

- Las **instrucciones** especificadas por el profesor, que el alumno deberá tener en cuenta antes de comenzar el juego.

Una vez que el alumno pulse el botón de "EMPEZAR", la información anterior desaparecerá de la pantalla y se mostrará la primera pregunta del juego así como el panel de información, que mostrará:

- **Nombre** del jugador.
- El **número de aciertos** del usuario entre los posibles, inicialmente cero.
- El **número de puntos** del usuario obtenidos entre los posibles, inicialmente cero. Cada pregunta puede tener diferente valor y puede restar, en caso de fallo, un porcentaje del valor anterior.
- **Nota** del alumno, que no deja de ser la puntuación obtenida en cada momento, ponderada sobre 10 puntos.

Al mismo tiempo, comenzará automáticamente la cuenta atrás que limita el tiempo para contestar dicha pregunta y se registrará en la base de datos el inicio de sesión de ese usuario en ese momento determinado. Ahora, tanto si se contesta la pregunta como si se agota el tiempo permitido para contestarla, la pregunta desaparecerá de la pantalla y se actualizará el panel de información superior. Así, se



indicará tras la corrección de la pregunta alguno de estos tres mensajes por pantalla:

- **Tiempo agotado:** en caso de que el jugador no haya contestado a ninguna pregunta y el contador haya alcanzado el valor cero.
- **Correcto:** en caso de que el jugador haya acertado la pregunta. Se mostrará junto con la puntuación obtenida correspondiente al valor de la pregunta.
- **Incorrecto:** en caso de que el jugador haya fallado la pregunta. Se mostrará junto con la puntuación a restar a causa del fallo. Será siempre un porcentaje del valor de la pregunta.

Las respuestas de cada cuestión siempre se presentarán en orden aleatorio. El resto de preguntas del juego seguirán el mismo procedimiento que la primera y al corregir cada pregunta, la **respuesta** dada por el alumno será **almacenada** en la base de datos con objeto de completar las estadísticas de la instancia del e-Liza.

Una vez contestada la última pregunta, se mostrará en pantalla el botón 'Volver al menú principal', que permitirá abandonar el juego, **cerrando** correctamente la sesión y registrándose la fecha y hora en que el jugador finaliza el juego. Si el jugador sale de la aplicación antes de terminar el juego, la sesión no se cerrará correctamente, lo que se verá reflejado posteriormente en las estadísticas.

Por otro lado, el contador viene acompañado de una barra horizontal que indica gráficamente el estado del mismo. La longitud de la barra disminuye linealmente de la misma forma en la que lo hace el contador. Asimismo, el color de la barra cambia linealmente de verde a rojo a medida que disminuye su anchura. Estos efectos se consiguen gracias a **CSS3** [23] utilizando la propiedad **transition**, que es compatible con los principales navegadores:

Todas las funciones javascript utilizadas para esta funcionalidad residen en el fichero **js/game.js**. Los aspectos de estilo pueden encontrarse en el fichero **css/style.css** y los ficheros PHP que son llamados por Ajax se encuentran en el directorio **php/game**. Esta organización hace sencillo editar cualquier característica de esta funcionalidad.



Administrar

Esta funcionalidad está orientada al uso exclusivo de la aplicación por parte del profesor con permisos de edición del curso. Abarca muchas otras funcionalidades del módulo relativas a la administración de la instancia de e-Liza v2. En primer lugar, la tarea más básica de esta sección consiste en ajustar la **configuración general** del juego acorde a lo deseado por el profesor. Así, los parámetros a editar son:

- **Nombre del juego**, que diferencia las distintas instancias de e-Liza en un curso.
- **Orden de las preguntas**: Aleatorio o Secuencial. El funcionamiento de este parámetro es detallado más abajo, en la sección Preguntas del juego.
- **Número máximo de intentos por persona**: Puede ser un número entero que haga de límite o ser cero o nulo, indicando que los intentos son ilimitados.
- **Número de preguntas** del e-Liza, que se corresponde con el campo *numberquestions* de la base de datos cuyo funcionamiento puede verse explicado en la sección 3.1 de este mismo capítulo. Podrá tomar, así, los valores 'Todas', 'Ilimitadas' o un número entero positivo.

En cuanto al resto de características disponibles, se abordan por separado los tres bloques que conforman la interfaz:

Banco de preguntas del curso

Una de las principales novedades del módulo en esta versión es la idea del **banco de preguntas**. Cada curso tiene asociado un conjunto de preguntas que se acumulan a medida que el profesor va introduciéndolas en la aplicación. Para todas las instancias de e-Liza v2 es posible seleccionar cualquier pregunta contenida en el banco de preguntas de su curso. De este modo, para nuevos juegos creados en el curso es posible **reutilizar** preguntas que fueron usadas en otros juegos del mismo curso.



Asimismo, cada curso también posee un **banco de etiquetas** que podrán ser utilizadas en cualquier instancia del curso para etiquetar las preguntas anteriores. Entiéndase el concepto de etiqueta como una cadena de caracteres que clasifica la pregunta con una determinada categoría de preguntas. Una pregunta puede estar asociada a muchas etiquetas y una etiqueta puede ser utilizada por muchas preguntas. Esta idea también resulta novedosa y pretende facilitar la organización de las preguntas dotándola de sofisticación.

La **versión anterior** de e-Liza carecía de estos conceptos y por ende no se permitía la reutilización de preguntas entre juegos, y la clasificación de las preguntas resultaba bastante más engorrosa al contar con categorías (una única asignable a cada pregunta), en lugar de etiquetas, donde cada pregunta puede asociarse a varias de las mismas.

En este bloque es posible gestionar completamente las preguntas del banco de preguntas así como de las etiquetas. Las funcionalidades de gestión se describen a continuación:

- **Lista de preguntas con buscador:** En el área superior se lista el conjunto de preguntas que conforman el banco de preguntas del curso. Viene acompañado de un buscador que actúa a medida que el usuario va escribiendo en la caja de texto y puede filtrar las preguntas por:
 - o **Título:** Muestra aquellas preguntas cuyo título contenga la cadena escrita en la caja de texto.
 - o **Contenido de la pregunta:** Muestra aquellas preguntas cuyo contenido (la pregunta en sí) contenga la cadena escrita en la caja de texto.
 - o **Etiquetas:** Muestra aquellas preguntas cuyas etiquetas asociadas contengan en su nombre la cadena escrita en la caja de texto.

Un detalle a tener en cuenta es que nada más cargarse la página de administración, el foco se centra automáticamente en el campo de entrada del buscador, situándose el cursor de escritura en dicho campo. Esto resulta cómodo para el usuario al permitirle



la búsqueda de preguntas introduciendo directamente la cadena de texto sin tener que situar el cursor manualmente en la caja de búsqueda. Esta funcionalidad es posible gracias a una de las nuevas características de **HTML5** para formularios, el atributo **autofocus** [24]. Asimismo, se incluyen dos características novedosas más propias de esta tecnología como son el atributo **placeholder** [25], que muestra un texto de ayuda en la caja de búsqueda y desaparece cuando el usuario comienza a escribir en ella, y el tipo **search** [26] como campo de entrada para formularios, que, a grandes rasgos, se comporta como un campo de texto normal pero añade un aspa para eliminar todo su contenido de una vez. La **compatibilidad** de estas características con los principales navegadores, a día de hoy, es la siguiente:

- Atributo **autofocus**:
- Atributo **placeholder**:
- Tipo de entrada **search**:
- **Lista de etiquetas:** En el área inferior del bloque se lista el conjunto de todas las etiquetas pertenecientes al curso. Al pulsar en el nombre de una de ellas se actualiza la lista anterior de preguntas, mostrando únicamente aquellas asociadas a dicha etiqueta.
- **Nueva pregunta:** Se muestra un panel frontal superpuesto a la página con un formulario a rellenar solicitando la información relativa a la nueva pregunta. Los datos a introducir son:
 - **Nombre** de la pregunta: sirve de identificación a la misma como nombre más corto que su propio contenido.
 - **Tipo:** especifica, como su propio nombre indica, el tipo de la pregunta, pudiendo ser:
 - **Genérica:** Se trata de una pregunta de tipo test habitual. Respuestas y pregunta en sí totalmente personalizables.
 - **Verdadero o falso:** Se asociarán automáticamente las palabras "Verdadero" o "Falso" como las respuestas a la pregunta.
 - **Es verdadero:** Se indica que la pregunta en sí toma la forma: "¿Cuál de las siguientes afirmaciones es verdadera?".



- **Es falso:** Se indica que la pregunta en sí toma la forma: "¿Cuál de las siguientes afirmaciones es falsa?".
- **Contenido de la pregunta:** donde se introduce el texto que conforma la pregunta en sí. Puede contener códigos de programación de distintos lenguajes sin ser interpretados, esto es, mostrados posteriormente en texto plano.
- **Respuestas:** permite añadir tantas respuestas como se desee a la pregunta, indicando para cada una de ellas, si se trata de una respuesta correcta o incorrecta.
- **Etiquetas:** permite asociar tantas etiquetas como se desee a la pregunta para tenerla completamente clasificada.
- **Tiempo para la pregunta:** se corresponde con el tiempo de que dispondrá el alumno para contestar dicha cuestión. Una cuenta regresiva se encargará de limitar este tiempo durante el juego.
- **Valor de la pregunta:** se corresponde con la puntuación obtenida por el alumno durante el juego al acertar dicha cuestión. Por defecto, las nuevas preguntas tienen un valor de 1 punto.
- **Penalización:** se trata de un número que indica el tanto por ciento del valor de la pregunta, que será restado a la puntuación del alumno al fallar dicha cuestión.
- **Editar pregunta:** Se muestra un panel frontal superpuesto a la página con un formulario a rellenar solicitando la información relativa a la pregunta cuya información se desea editar. Los datos a introducir son los mismos que para las nuevas preguntas.
- **Eliminar pregunta:** Permite eliminar permanentemente una pregunta del banco de preguntas del curso y por tanto, de la base de datos y de la plataforma. También quedarán borrados, por consiguiente, los datos almacenados que contengan alguna relación con la pregunta anterior, como sus respuestas. Se muestra un panel frontal superpuesto a la página con un mensaje informativo de confirmación que lleva a ejecutar la acción.
- **Agregar pregunta al juego:** Esta será la forma de operar para seleccionar las preguntas pertenecientes al banco integrándolas



en el juego correspondiente a la instancia concreta del e-Liza. Cada pregunta es seleccionable una sola vez para cada juego y quedará perfectamente indicado en la lista si una cuestión ya ha sido agregada o por el contrario, está permitido añadirla.

Una característica muy interesante que nos brinda de nuevo **HTML5**, es la posibilidad de arrastrar y soltar (**Drag&Drop**) las cuestiones del banco de preguntas en el área de preguntas del juego, agregando del mismo modo la pregunta seleccionada a esta lista. Así, esto es un buen ejemplo para manifestar que se van completando las funcionalidades empleando las últimas tecnologías web.

- **Ver pregunta:** Al pulsar sobre el nombre de cualquier pregunta se muestra un panel frontal superpuesto a la página con toda la información relativa a ella, del mismo modo que lo hace la funcionalidad *editar pregunta* pero sin la posibilidad de edición de los datos.
- **Gestionar etiquetas:** Como su propio nombre indica, esta funcionalidad permite administrar el conjunto de etiquetas pertenecientes a un curso visualizando la lista de todas ellas y posibilitando:
 - o Crear nueva etiqueta
 - o Editar etiqueta existente
 - o Eliminar etiqueta de forma permanente

Preguntas del juego

El bloque derecho de la interfaz simplemente contiene la lista de todas las preguntas que han sido seleccionadas para conformar el juego. Tienen un **orden** específico indicado por un número correspondiente al **peso** de la pregunta. Las preguntas con menor peso aparecerán en la lista antes que las preguntas con mayor peso. Personalizar este orden resulta útil cuando está seleccionado el parámetro del juego 'orden de las preguntas' con valor 'Secuencial', de modo que durante el juego las cuestiones aparecerán en el orden indicado por los pesos. Si el parámetro del juego 'orden de las preguntas' tiene el valor 'Aleatorio', entonces



esos pesos carecen de sentido y las preguntas aparecerán evidentemente en orden aleatorio.

Así, sobre cada una de las preguntas de la lista, es posible:

- **Ver pregunta:** Al pulsar sobre el nombre de cualquier pregunta se muestra un panel frontal superpuesto a la página con toda la información relativa a ella.
- **Eliminar pregunta:** Permite eliminar la pregunta de la selección de preguntas del juego y no del banco de preguntas del curso, donde permanecería intacta.
- **Incrementar el peso de la pregunta:** intercambiando el peso de la pregunta por el de la pregunta de peso una unidad mayor.
- **Decrecer el peso de la pregunta:** intercambiando el peso de la pregunta por el de la pregunta de peso una unidad menor.

Preguntas propuestas por los alumnos

El último bloque de la página de administración de la instancia del e-Liza v2 contiene la lista de todas las preguntas que han sido propuestas por los alumnos a lo largo de todo el curso con objeto de ser añadidas al banco de preguntas. En esta tabla es posible visualizar:

- La **fecha** en que fue sugerida la pregunta.
- El **nombre del usuario** que ha realizado dicha proposición.
- El **nombre de la pregunta** que, tras ser pulsado, despliega un panel frontal superpuesto a la página con toda la información relativa a la cuestión.
- El **tipo de pregunta** seleccionado en su creación (Véase la sección 2.1).
- El **número de respuestas** asociadas.
- El **estado** en que se encuentra la propuesta, que puede ser:
 - o **Pendiente:** la pregunta aún no ha sido revisada por el profesor.



- **Aprobado:** la pregunta ha sido aprobada por el profesor y añadida correctamente al banco de preguntas del curso, donde puede ser reeditada.
- **Rechazado:** la pregunta ha sido rechazada por el profesor.

Como funcionalidades adheridas a esta tabla, se describen:

- **Aprobar propuesta:** El profesor da el visto bueno a la sugerencia del alumno integrándola en el banco de preguntas del curso y cambiando el estado de la propuesta a 'Aprobado'.
- **Rechazar propuesta:** El profesor no da el visto bueno a la sugerencia del alumno, que queda marcada con el estado 'Rechazado'.
- **Eliminar propuesta:** El profesor borra directamente de la base de datos la pregunta propuesta por el alumno junto con sus respuestas asociadas. Si una de dichas preguntas posee el estado 'Aprobado' y es eliminada, no se verá afectada la copia que se hizo sobre el banco de preguntas.

Cabe mencionar que, para conseguir cambiar el contenido de la página ocultando y mostrando áreas de la misma, exponiendo nuevos paneles o actualizando las listas de preguntas sin dirigirse a nuevas páginas, se ha utilizado de nuevo la tecnología **AJAX** [Anexo 3].

Todas las funciones javascript utilizadas para la administración residen en el fichero **js/admin.js**. Los aspectos de estilo pueden encontrarse en el fichero **css/style.css** y los ficheros PHP que son llamados por Ajax se encuentran en el directorio **php/admin**. Esta organización hace sencillo editar cualquier característica de esta funcionalidad global.

Proponer preguntas

Esta funcionalidad está orientada al uso de la aplicación por parte del alumno y diseñada con el objetivo de constituir una herramienta para la gestión de sugerencias de nuevas preguntas para el juego.



- **Lista de las propuestas:** Presenta en forma de tabla cada una de las propuestas asociadas al curso creadas por el usuario identificado en la plataforma. Muestra la siguiente información:
 - o **Fecha** de creación de la propuesta.
 - o **Nombre** de la pregunta propuesta, que, tras ser pulsado, despliega un panel frontal superpuesto a la página con toda la información relativa a la cuestión.
 - o **Tipo** de la pregunta (Véase la sección 2.1).
 - o **Número de respuestas** asociadas.
 - o **Estado** de la propuesta (Véase la sección 2.3).
- **Nueva propuesta:** Se muestra un panel frontal superpuesto a la página con un formulario a rellenar solicitando la información relativa a la nueva propuesta de pregunta. Los datos a introducir son los mismos que se especifican en la sección 2.1 para nuevas preguntas, salvo el valor de la misma y su porcentaje de penalización, que serán completados por el profesor en caso de aprobación de la cuestión.
- **Editar propuesta:** Se muestra un panel frontal superpuesto a la página con un formulario a rellenar solicitando la información relativa a la propuesta cuya información se desea editar. Los datos a introducir son los mismos que para las nuevas propuestas.
- **Eliminar propuesta:** Permite eliminar permanentemente una propuesta de la lista anterior y por tanto, de la base de datos y la plataforma. También quedarán borradas las respuestas asociadas. Se muestra un panel frontal superpuesto a la página con un mensaje informativo de confirmación que lleva a ejecutar la acción.

Todas las funciones javascript utilizadas para esta sección residen en el fichero **js/suggestions.js**. Los aspectos de estilo pueden encontrarse en el fichero **css/style.css** y los ficheros PHP que son llamados por Ajax se encuentran en el directorio **php/suggestions**. Esta organización hace sencillo editar cualquier característica de esta funcionalidad global.



Estadísticas

Esta funcionalidad está orientada al uso de la aplicación tanto por parte del alumno como del profesor, pero ofrece diferentes resultados a cada uno de estos roles. De este modo, al profesor se le presenta un **análisis completo de todas las estadísticas** de esa instancia del e-Liza, incluyendo las de todos los usuarios que han participado en el juego, mientras que al alumno tan sólo se le mostrarán sus **estadísticas personales**. El desarrollo de esta parte del módulo ha supuesto un buen ejercicio para la práctica del lenguaje **SQL** requiriendo algunas consultas en la base de datos de carácter un tanto avanzado.

Estadísticas completas

Una de las características esenciales de esta segunda versión de e-Liza es un control tan detallado como intuitivo de las estadísticas. Esta sección puede dividirse en cuatro apartados que están, en todo momento, a disposición del profesor del curso:

Estadísticas globales

Se refiere al conjunto de datos registrados por la aplicación cuando algún alumno del curso juega una partida en la instancia dada del módulo en la que se encuentra el profesor. De esta manera, se presentan las siguientes estadísticas relativas al conjunto global de los alumnos del curso para ese juego:

- **Aciertos:** Se muestra el número de aciertos de las respuestas de los alumnos así como el porcentaje de ellos con respecto a las preguntas totales aparecidas.
- **Fallos:** Se muestra el número de fallos de las respuestas de los alumnos así como el porcentaje de ellos con respecto a las preguntas totales aparecidas.
- **Sin contestar:** Se muestra el número de respuestas de los alumnos que no han sido contestadas, bien porque se les agotó el tiempo, bien porque decidieron no contestarlas, así



como el porcentaje de ellas con respecto a las preguntas totales aparecidas.

- **Número de sesiones:** Indica el número de veces que se ha jugado a esa instancia del módulo.
- **Número medio de sesiones por usuario:** Indica el número medio de veces por usuario que se ha jugado a esa instancia del módulo (Número de sesiones / número de usuarios que han participado).
- **Número de sesiones sin cerrar:** Indica el número de veces que se ha jugado a esa instancia del módulo en las que el jugador no ha terminado la partida.
- **Tiempo medio de respuesta:** Indica el tiempo medio en segundos que el conjunto de alumnos del curso ha tardado en contestar.
- **Nota media:** Indica la nota media obtenida por el conjunto de los alumnos del curso en las sesiones llevadas a cabo para esta instancia del e-Liza.
- **Nota máxima:** Indica la nota máxima obtenida por el conjunto de los alumnos del curso en las sesiones llevadas a cabo para esta instancia del e-Liza.

Además, figuran dos imágenes que dotan de carácter gráfico a la sección de estadísticas. Para su generación se ha utilizado la librería pChart [16].

- **Gráfica circular o de tarta:** que muestra de forma más representativa el porcentaje de aciertos con respecto al de fallos y respuestas sin contestar.



Ilustración 9: Gráfica circular o de tarta

- **Gráfica temporal:** que muestra la nota media del conjunto total de alumnos del curso, agrupada por meses, en el rango de fechas en que aquellos han participado del juego.

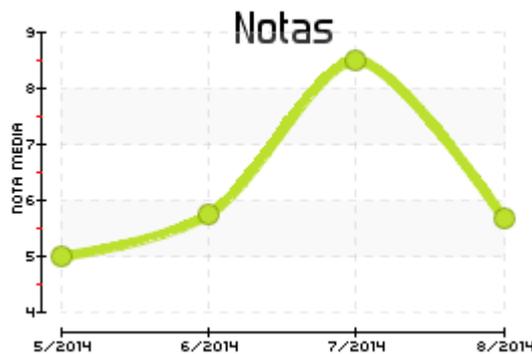


Ilustración 10: Gráfica temporal

Resulta interesante considerar el algoritmo empleado para el cálculo de los datos de esta gráfica, donde el *array* \$data contiene el valor o la altura de cada punto de la misma y el *array* \$labels contiene las etiquetas del eje de abscisas [Anexo 7].

Estadísticas por usuario

Esta tarea de la aplicación se encarga de exponer, en forma de tabla, la lista de todos los usuarios que han participado alguna vez en el juego. Para cada uno de ellos se calcula y se muestra:



- La **nota media** de todas las sesiones que haya llevado a cabo en esa instancia del módulo.
- La **mejor nota** de todas las sesiones que haya llevado a cabo en esa instancia del módulo.
- El **número de sesiones** que haya llevado a cabo en esa instancia del módulo.
- El **número de sesiones sin cerrar** en esa instancia del módulo, esto es, el número de juegos comenzados por el usuario que finalmente no han alcanzado su término al haber salido de la aplicación antes acabar de contestar todas las preguntas.
- El **tiempo medio de respuesta**, como tiempo medio que tarda en contestar el usuario al conjunto de todas las preguntas que han aparecido durante todas sus sesiones asociadas en esa instancia del módulo.

Resulta interesante comentar la posibilidad de **ordenar la tabla** por cualquiera de los campos anteriores, tanto ascendente como descendentemente.

Además, es posible **ampliar** la visualización de la **información** relativa a cada usuario pulsando directamente sobre su nombre en la tabla. En este momento, se despliega un panel frontal superpuesto a la página que, además de los datos anteriores, muestra el número de **aciertos, fallos y preguntas sin contestar**, junto con sus respectivos porcentajes, registrados en las respuestas que da el usuario durante los juegos. Al igual que en el apartado anterior, se dibuja una **gráfica circular** representando dichos porcentajes. Y junto con toda esta información, se presenta en forma de tabla, de nuevo, la **lista de las sesiones** llevadas a cabo en esa instancia del e-Liza v2 por el usuario seleccionado.

Estadísticas por sesiones

Este apartado lista, en forma de tabla, todas las sesiones habidas para esa instancia del e-Liza v2. Para cada una se muestran:



- El **nombre del usuario** que llevó a cabo la sesión. Como en el apartado anterior, es posible ampliar su información asociada pulsando simplemente sobre su nombre.
- La **fecha de apertura** de la sesión, que indica la fecha y hora en que fue iniciado el juego.
- La **fecha de cierre** de la sesión, que indica la fecha y hora en que fue terminado el juego. En el caso en el que la sesión no fuera cerrada, se indica en este mismo campo.
- La **dirección IP** del ordenador desde el que el usuario efectuó la partida.
- El **tiempo medio de respuesta** de todas las preguntas que conformaron la sesión.
- La **nota** sobre 10 obtenida durante la sesión.

Resulta interesante comentar la posibilidad de **ordenar** la tabla por cualquiera de los campos anteriores, tanto ascendente como descendentemente.

Además, es posible **ampliar** la visualización de la **información** relativa a cada sesión. De este modo, se despliega un panel frontal superpuesto a la página que, además de los datos anteriores, muestra el número de **aciertos**, **fallos** y **preguntas sin contestar**, junto con sus respectivos porcentajes, registrados en las respuestas que da el usuario durante la sesión seleccionada. Al igual que en el apartado anterior, se dibuja una **gráfica circular** representando dichos porcentajes. Y junto con toda esta información, se presenta en forma de tabla, de nuevo, la **lista de respuestas** que dio el usuario durante la sesión a cada una de las preguntas aparecidas, indicando la pregunta a la que responde, su valor, su porcentaje de penalización, el tiempo que tomó el usuario en contestarla y si es o no correcta.

Estadísticas por preguntas

Esta funcionalidad puede ser interesante para el profesor. Se presenta la lista de todas las preguntas que han aparecido durante los juegos de esa instancia del módulo, mostrándose para cada una de ellas:

- El número de **veces que ha aparecido**.
- El número de **veces que ha sido acertada**, y su respectivo porcentaje.
- El número de **veces que ha sido fallada**, y su respectivo porcentaje.
- El número de **veces que no ha sido contestada**.

Una posible **utilidad** de esto es que el profesor observe las preguntas con alto o bajo porcentaje de acierto para obtener algunas conclusiones interesantes. Por ejemplo, si una pregunta tiene un alto porcentaje de fallos puede ser que sea por estar mal formulada. Si una pregunta tiene un alto porcentaje de veces sin ser contestada puede ser debido a que la pregunta es demasiado complicada. Y sin embargo, si tiene un excesivo porcentaje de aciertos, puede resultar que la pregunta sea demasiado fácil con respecto a lo deseado.

Resulta interesante comentar la posibilidad de **ordenar** la tabla por cualquiera de los campos anteriores tanto ascendente como descendentemente.

Además, es posible **visualizar el contenido** en sí de cada pregunta. De este modo, se despliega un panel frontal superpuesto a la página que presenta el nombre de la pregunta, su tipo, sus respuestas asociadas y el resto de información que la detalla.

Estadísticas personales

Es evidente que cada alumno solo tendrá permiso para contemplar sus propias estadísticas y no las del resto de usuarios, por temas de privacidad. De esta manera, cada alumno del curso únicamente visualizará los datos asociados a su identificador de usuario:

- **Aciertos:** Se muestra el número de aciertos de las respuestas del alumno así como el porcentaje de ellos con respecto a las preguntas totales aparecidas en todas sus partidas de la instancia del módulo.



- **Fallos:** Se muestra el número de fallos de las respuestas del alumno así como el porcentaje de ellos con respecto a las preguntas totales aparecidas en todas sus partidas de la instancia del módulo.
- **Sin contestar:** Se muestra el número de respuestas del alumno que no han sido contestadas, bien porque se le agotó el tiempo, bien porque, simplemente, decidió no contestarlas, así como el porcentaje de ellas con respecto a las preguntas totales aparecidas en todas sus partidas de la instancia del módulo.
- **Número de sesiones:** Indica el número de veces que el alumno ha jugado a esa instancia del módulo.
- **Número de sesiones sin cerrar:** Indica el número de veces que el alumno ha jugado a esa instancia del módulo en las que no ha terminado la partida.
- **Tiempo medio de respuesta:** Indica el tiempo medio en segundos que el alumno ha tardado en contestar.
- **Nota media:** Indica la nota media obtenida por el alumno en las sesiones llevadas a cabo para esta instancia del e-Liza.
- **Nota máxima:** Indica la nota máxima obtenida por el alumno en las sesiones llevadas a cabo para esta instancia del e-Liza.

Además, se muestra la **lista** de todas las **sesiones** correspondientes al alumno en esa instancia de e-Liza v2, y es posible **ampliar** la visualización de la **información** relativa a cada sesión, del mismo modo que para los apartados anteriores.

Todas las funciones javascript utilizadas para las estadísticas residen en el fichero **js/statistics.js**. Los aspectos de estilo pueden encontrarse en el fichero **css/style.css** y los ficheros PHP que son llamados por Ajax se encuentran en el directorio **php/statistics**. Esta organización hace sencillo editar cualquier característica de esta funcionalidad global.

Jugar en grupo

Esta sección ha sido diseñada para permitir el juego en grupos de alumnos, de modo que cualquier usuario puede crear uno de estos



grupos e integrar a más alumnos al grupo, retándolos a superar su nota obtenida a una única partida.

En primer lugar, se expondrá la **lista** de todas las partidas en grupo de la instancia del e-Liza v2 accedida en las que el alumno participa, bien porque haya sido retado por otro alumno, bien porque precisamente él es el creador. Para cada una de ellas, el alumno podrá jugar una partida donde obtendrá una nota que competirá con las de sus contrincantes.

Pero la funcionalidad esencial aquí es la de **crear nueva partida en grupo**. Se muestra un panel frontal superpuesto a la página con un formulario a rellenar solicitando la información relativa a la partida. Los datos a cumplimentar son:

- **Nombre de la partida en grupo**, a elegir por el creador.
- **Conjunto de usuarios** que son retados, a seleccionar entre la lista de todos los alumnos matriculados en el curso.

Es interesante considerar el código utilizado para obtener la lista de usuarios matriculados en el curso, que combina lenguaje PHP con algunas consultas SQL. [**Anexo 8**]

En la **versión anterior** del e-Liza, ésta no era la forma de operar para formar los grupos de alumnos que competirían entre sí, sino que una partida inicialmente creada por un alumno estaba asociada a una contraseña que posteriormente usarían otros alumnos para unirse al grupo. El nuevo procedimiento que ofrece e-Liza v2 evita el uso de contraseñas y los grupos quedan completamente formados en el momento de su creación. Además, añade la privacidad de las partidas en grupo, de modo que un alumno no sepa de la existencia de aquellas en las que no participe.

En el momento en que un alumno es invitado a pertenecer a un grupo, se le permite un sólo intento de juego para conseguir la máxima nota posible con la que competirá con el resto de los integrantes del grupo. Una vez que todos ellos hayan defendido sus conocimientos, se decidirá el ganador de la partida en grupo.



Todas las funciones javascript utilizadas para la administración residen en el fichero **js/admin.js**. Los aspectos de estilo pueden encontrarse en el fichero **css/style.css** y los ficheros PHP que son llamados por Ajax se encuentran en el directorio **php/admin**. Esta organización hace sencillo editar cualquier característica de esta funcionalidad global.

Por terminar con un comentario concerniente a todas las funcionalidades, cabe mencionar que para conseguir cambiar el contenido de la página ocultando y mostrando áreas de la misma, exponiendo nuevos paneles o actualizando el contenido de algunos bloques, sin dirigirse a nuevas páginas, se ha utilizado la tecnología **AJAX** [Anexo 3]. Distintas funciones javascript realizan peticiones al servidor de ciertos ficheros PHP que, a su vez, consultan la base de datos y generan el nuevo contenido a mostrar en alguna región de la página. Gran parte del módulo e-Liza v2 ha sido desarrollado haciendo uso de esta tecnología.

Puede verse gráficamente, a modo aclaratorio, un estudio global del comportamiento del sistema en el **diagrama de casos de uso** del [Anexo 9]. Además, para cada una de las funcionalidades expuestas, se presenta en el [Anexo 10] un **diagrama de actividades UML** que dibuja el flujo del programa en sus distintas secciones.

Capítulo 6. Manual de usuario

El siguiente breviarío pretender servir de guía para el uso de la aplicación. En función del rol ejercido en la plataforma por el usuario identificado, interesa aprender el manejo de unas partes del programa u otras, dividiendo el capítulo en Manual para el profesor y Manual para el alumno.

Manual para el profesor

Instalación del módulo

Existe un procedimiento común para la instalación de cualquier complemento sobre la plataforma Moodle: situar los ficheros del módulo en la localización apropiada (directorio **/mod** para los módulos de actividad) y visitar la página de administración principal *Administración > Administración del sitio > Notificaciones*.

Existen **dos formas** de hacer esto. La primera forma parte de las nuevas características de Moodle 2.5 en adelante y exige que el proceso del servidor web tenga acceso de escritura a la carpeta */mod*. Y la segunda conforma el método tradicional. A continuación se explican ambas:

- **Instalación mediante archivo ZIP subido al sitio**
 - o Acceda a su sitio Moodle como administrador y vaya a *Administración > Administración del sitio > Extensiones > Instalar módulos externos*.
 - o Suba el paquete ZIP, seleccione el tipo de extensión: Módulo (mod), acepte la casilla de Reconocimiento y después pulse el botón 'Instalar módulo externo desde un archivo ZIP'.

▼ Instalar módulo externo desde un archivo ZIP ⓘ

Tipo de extensión* ⓘ Módulo (mod) ▼

✓ Plugin type location C:\xampp\htdocs\moodle/mod is writable

Paquete ZIP* ⓘ Seleccione un archivo...

eliza.zip

Reconocimiento* Entiendo que es mi responsabilidad disponer de copias de seguridad completas de este sitio antes de instalar módulos externos (ad-ons). Acepto y entiendo que los módulos externos (especialmente aquellos procedentes de fuentes no oficiales) pueden contener agujeros de seguridad, hacer que el sitio quede fuera de servicio, o provocar fugas de datos privados o la pérdida de los mismos.

[Ver más...](#)

Instalar módulo externo desde archivo ZIP

Ilustración 11: Instalación mediante archivo ZIP

- Revise que aparezca el mensaje de que pasó la validación (Validation passed!) y después pulse el botón *Instalar el complemento (Install add-on)*.
 - Usted será automáticamente redirigido hacia *Administración > Administración del sitio > Notificaciones* y tendrá lugar el proceso de instalación.
- **Instalación manual en el servidor**
- Copie manualmente el directorio *eliza* correspondiente a módulo en el directorio */mod* de la instalación de Moodle.

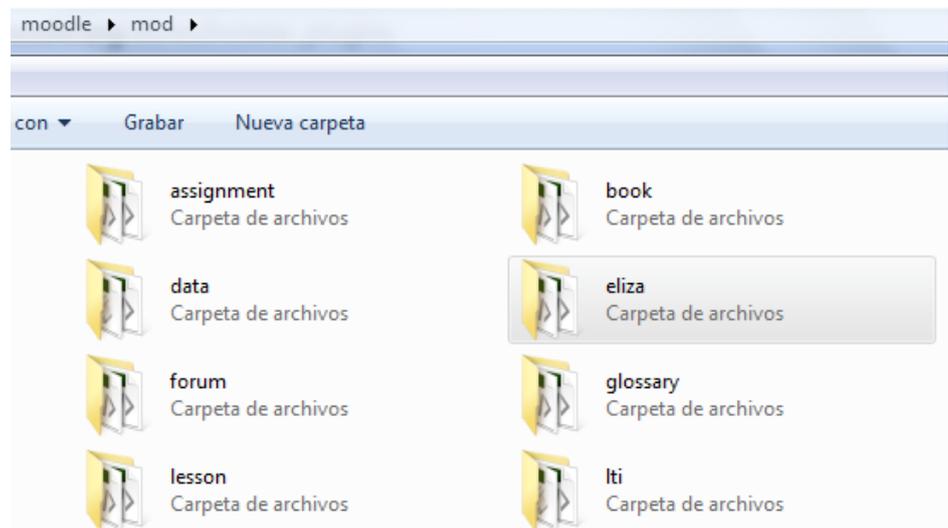


Ilustración 12: Instalación manual en el servidor

- Acceda a su sitio Moodle como administrador y vaya a *Administración > Administración del sitio > Notificaciones*.

Desinstalación del módulo

- Vaya a *Administración > Administración del sitio > Extensiones > Vista general de extensiones* y haga clic al botón de *Desinstalar* en la fila de la tabla correspondiente a *eliza*.

 Consulta mod_choice	Estándar	2014051200	Habilitado	Desinstalar
 Base de datos mod_data	Estándar	2014051200	Habilitado	Configuración
 eliza mod_eliza	Módulo externo	2014070602	Habilitado	Desinstalar
 Módulo de encuesta mod_feedback	Estándar	2014051200	Deshabilitado	Configuración
 Carpeta mod_folder	Estándar	2014051200	Habilitado	Configuración Desinstalar

Ilustración 13: Desinstalación del módulo

- Confirme la desinstalación.

Creación de la actividad

- Tras acceder en la plataforma con la cuenta de profesor, acceda al curso deseado y active su edición. Para ello, haga clic en el botón *Activar edición*.
- Pulse en *'+Añadir una actividad o un recurso'* en la semana donde quiere crear la instancia.
- Seleccione la actividad *eliza* y haga clic en *Agregar*.

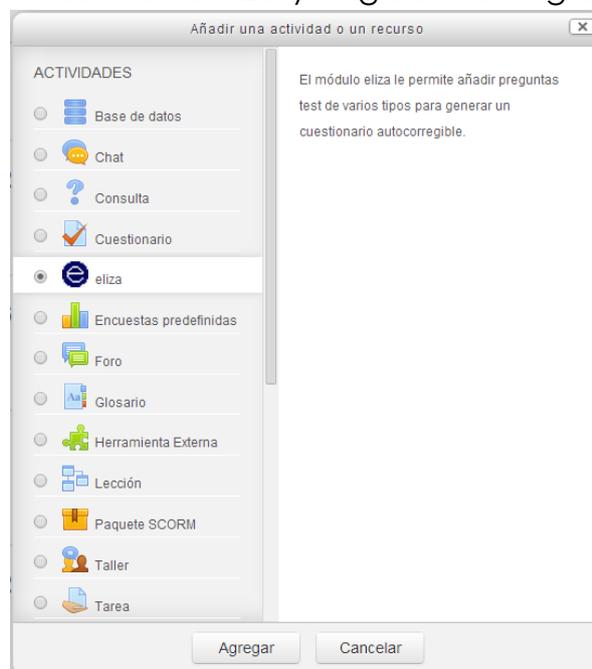


Ilustración 14: Creación de la actividad

- Aparecerá un formulario solicitando los parámetros del juego donde deberá rellenarse:
 - o Nombre del juego
 - o Descripción
 - o Orden de las preguntas
 - o Número de intentos máximos permitidos por usuario
 - o Número de preguntas del juego

Menú principal



Ilustración 15: Menú principal del programa para el profesor

Se presentan 5 botones para el acceso a las distintas secciones de la aplicación. La barra horizontal inferior contiene el **nombre del juego** creado, en el ejemplo de la imagen: 'Bloque II | FOSO'. Al pasar el cursor por cada uno de los botones, en esta barra aparecerá el título de la sección asociada a la imagen del botón.

Aunque el rol de profesor puede acceder a todas las secciones del programa, incluyendo la de juego o proposición de preguntas como si de un alumno más se tratase, sólo resulta interesante analizar los apartados Administrar y Estadísticas.

Administrar



Esta sección permite efectuar cualquier cambio relativo al juego.

Configuración de los parámetros generales del juego

Si pulsa en el botón 'Configuración general', se mostrará un panel frontal superpuesto a la página con un formulario a rellenar solicitando los parámetros principales del juego, que coinciden con cuatro de los anteriores explicados para la creación de la actividad.



Ilustración 16: Configuración general del juego

Gestión del banco de preguntas

El bloque izquierdo de la página de administración lista el conjunto de todas las preguntas existentes en un curso. Para cada una de ellas es posible editar su contenido o su eliminación del banco.

Se incluye un **buscador** que actúa a medida que el profesor va escribiendo y filtra las preguntas en función de su título, contenido (la pregunta en sí) y/o nombre de sus etiquetas asociadas.

Incluir en la búsqueda: Nueva pregunta

Título Contenido de la pregunta Etiquetas

Busca la pregunta

Ilustración 17: buscador del banco de preguntas

Para añadir una **nueva pregunta** al banco tan sólo pulse el botón 'Nueva pregunta' y complete el formulario que aparece, cuyos campos a rellenar son:



Ilustración 18: Nueva pregunta

- **Nombre** de la pregunta, que será su identificador en las listas de preguntas.
- **Tipo** de la pregunta, que puede ser:
 - **Genérica:** Se trata de una pregunta de tipo test habitual. Respuestas y pregunta en sí son totalmente personalizables.
 - **Verdadero o falso:** Se asociarán automáticamente las palabras "Verdadero" o "Falso" como las respuestas a la pregunta.

- **Es verdadero:** Se indica que la pregunta en sí toma la forma: “¿Cuál de las siguientes afirmaciones es verdadera?”.
- **Es falso:** Se indica que la pregunta en sí toma la forma: “¿Cuál de las siguientes afirmaciones es falsa?”.
- **Contenido de la pregunta:** donde se introduce el texto que conforma la pregunta en sí. Puede contener códigos de programación que se mostrarán como texto plano.
- **Respuestas:** permite añadir tantas respuestas como se desee a la pregunta indicando para cada una de ellas si se trata de una respuesta correcta o incorrecta.



Ilustración 19: Nueva respuesta

- **Etiquetas:** permite asociar tantas etiquetas como se desee a la pregunta para tenerla completamente clasificada. Deben haber sido previamente creadas.
- **Tiempo** para la pregunta: se corresponde con el tiempo de que dispondrá el alumno para contestar dicha cuestión. Una cuenta regresiva se encargará de limitar este tiempo durante el juego.



- **Valor** de la pregunta: se corresponde con la puntuación obtenida por el alumno durante el juego al acertar dicha cuestión. Por defecto, las nuevas preguntas tienen un valor de 1 punto.
- **Penalización**: se trata de un número que indica el tanto por ciento del valor de la pregunta que será restado a la puntuación del alumno al fallar dicha cuestión.

Gestión de las etiquetas

La parte inferior del bloque del banco de preguntas contiene la lista de todas las etiquetas disponibles en un curso. Al pulsar sobre el nombre de alguna de ellas, la lista de preguntas del banco se verá filtrada por aquellas que estén asociadas a dicha etiqueta.

Al hacer clic en 'Gestionar etiquetas' podrá ver la lista de todas las etiquetas del curso en forma de tabla. Y para cada una de ellas puede editar su nombre o llevar a cabo su borrado. Para añadir una nueva etiqueta simplemente haga clic en 'Añadir etiqueta' y escriba el nombre deseado.

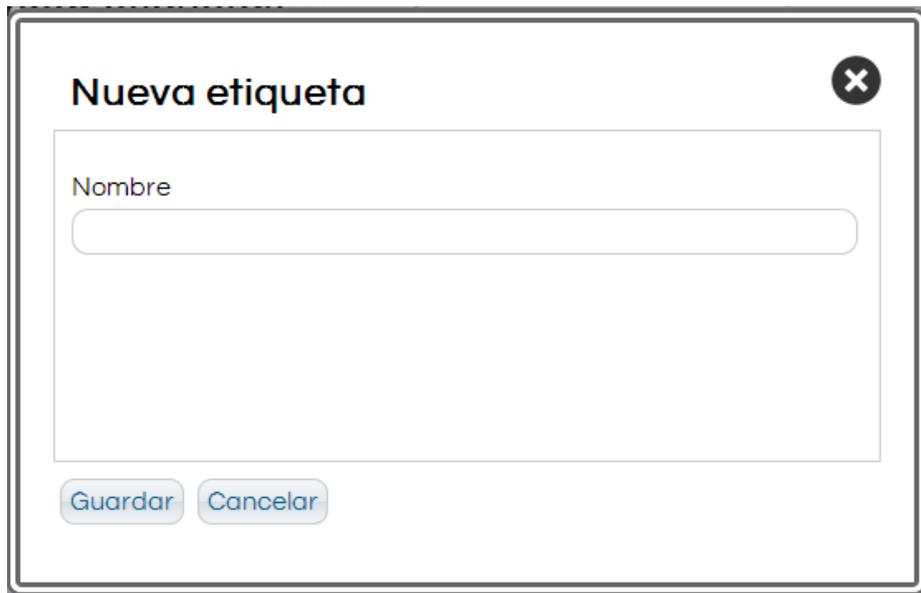


Ilustración 20: Nueva etiqueta

Gestión de preguntas del juego

El bloque de la derecha de la página de administración reúne el conjunto de preguntas que conformarán el juego. Todas ellas han sido agregadas del banco de preguntas del curso; por tanto, si se desea añadir una nueva pregunta, en primer lugar deberá crear la pregunta en él y posteriormente seleccionar dicha pregunta para que forme parte del juego.

Para agregar una pregunta del banco a la lista de preguntas del juego, pulse en 'Agregar', en la fila de la cuestión asociada. También puede arrastrar directamente con el ratón la pregunta y soltarla en el bloque de la derecha.

Un detalle más a tener en cuenta es el orden que toman las preguntas en la lista de preguntas del juego, puesto que ese será finalmente el orden real en el que aparezcan, suponiendo que el parámetro general 'Orden de las preguntas' tome el valor 'Secuencial'. Es posible aumentar

o disminuir una posición a una pregunta pulsando sobre las flechas que la acompañan:



Ilustración 21: Flechas de ordenación de las preguntas

Gestión de las preguntas propuestas por los alumnos

El bloque inferior de la página de administración lista todas las propuestas de preguntas que los alumnos han sugerido en ese curso. Cada una de ellas tiene asignado un estado, que puede ser:

- **Pendiente**, lo que indica que la propuesta está a la espera de aprobación por el profesor.
- **Aprobado**, lo que indica que la propuesta ha sido aceptada por el profesor.
- **Rechazado**, lo que indica que el profesor ha rechazado la propuesta del alumno.

Para ver toda la información de la pregunta propuesta por el alumno, basta con pulsar sobre su nombre. Puede además cambiar el estado de la propuesta **aprobándola** o **rechazándola**. En caso de aprobarla, su contenido será copiado al banco de preguntas del curso e inmediatamente editado, para que el profesor asigne el valor de la pregunta y su penalización así como para que edite algún aspecto de la misma, si fuese necesario.

Estado	
PENDIENTE	eliminar aprobar rechazar
APROBADO	eliminar
APROBADO	eliminar
APROBADO	eliminar
RECHAZADO	eliminar aprobar

Ilustración 22: Ejemplo de estados para preguntas propuestas

También es posible eliminar completamente el registro de la propuesta en caso de no querer almacenarla.

Ver estadísticas completas



Esta sección genera un completo análisis de todos los movimientos de los alumnos en el juego. Se presentan cuatro áreas que reflejan:

- Estadísticas globales del juego
- Estadísticas por usuario
- Estadísticas por sesión (cada vez que un alumno juega)
- Estadísticas por pregunta

Para ver una información más detallada de las estadísticas asociadas a un alumno, tan sólo haga clic sobre su nombre. Para ver una información más detallada de una sesión, pulse en 'detalles'. Y para ver el contenido de una pregunta haga clic sobre su nombre también.

Manual para el alumno

Menú principal



Ilustración 23: Menú principal del programa para el alumno

Se presentan cuatro botones para el acceso a las distintas secciones de la aplicación. La barra horizontal inferior contiene el **nombre del juego** creado, en el ejemplo de la imagen: 'Bloque II | FOSO'. Al pasar el cursor por cada uno de los botones, en esta barra aparecerá el título de la sección asociada a la imagen del botón.

Jugar



En primera instancia se muestran en una tabla los **parámetros del juego**, que previamente han sido definidos por el profesor. A mayores de su nombre de usuario, en la plataforma se muestran:

- El **orden de las preguntas**, que puede ser aleatorio o secuencial, indicando así que la aparición de las preguntas del juego seguirá tal orden.
- El **número de intentos** empleados en jugar correspondientes al número de partidas jugadas por usted así como el número máximo de intentos totales que tiene disponibles.



- El **número de preguntas** que compondrán el juego. Si el número de preguntas es ilimitado, podrá abandonar el juego tras contestar cualquiera de ellas.
- Las **instrucciones** especificadas por el profesor, que deberá tener en cuenta antes de comenzar el juego.

Una vez pulse el botón "EMPEZAR", se registrará el inicio del juego y deberá completar todas las preguntas antes de salir de la aplicación; requisito indispensable para que la partida se cierre correctamente y tenga validez. De lo contrario, quedará reflejada en las estadísticas esta situación.

Para contestar a las preguntas simplemente pulse sobre la respuesta elegida. En el área superior de la página de juego se expone un **panel de información** que mostrará:

- **Nombre** del jugador.
- El **número de aciertos** entre los posibles.
- El **número de puntos** obtenidos entre los posibles.
- **Nota**, que no deja de ser la puntuación obtenida en cada momento ponderada sobre 10 puntos.



Ilustración 24: Panel de información del juego

En el área inferior de la página se observa una **cuenta regresiva** que limita el tiempo para contestar la pregunta.

Y en la esquina superior derecha del área de la pregunta y sus respuestas, se muestra el **valor de la pregunta**, con fondo verde, así como el **porcentaje de penalización**, con fondo rojo.

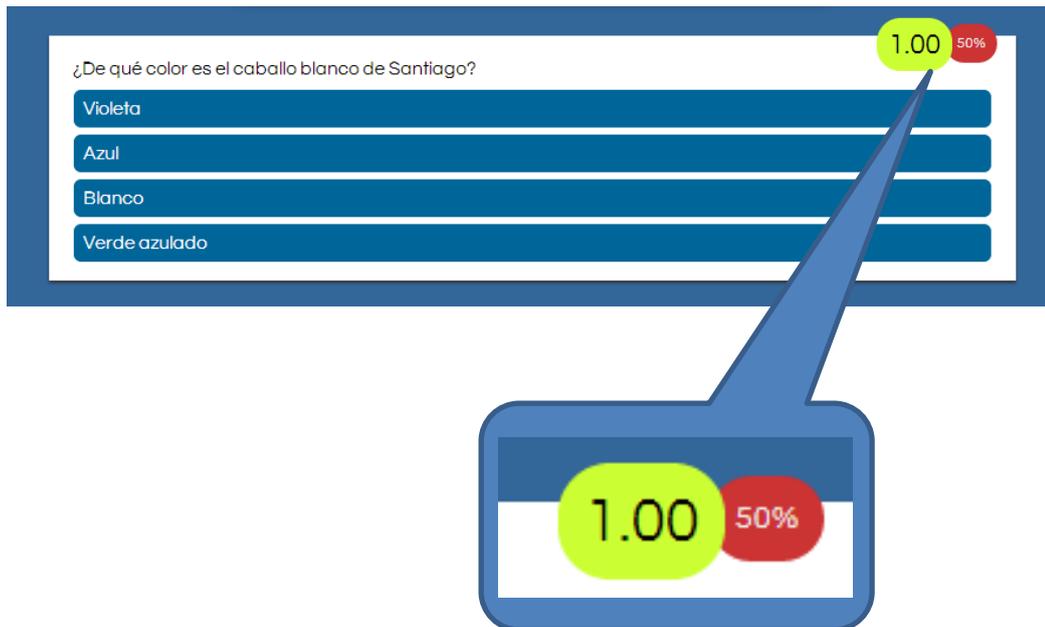


Ilustración 25: Valor de la pregunta y porcentaje de penalización

Si usted acierta la respuesta, se sumará el valor de la pregunta a la puntuación del juego. Si falla, se restará el porcentaje de penalización definido sobre el valor de la cuestión. Si no contesta, bien porque ha pasado de la pregunta, bien porque se ha agotado el tiempo para contestarla, la puntuación del juego permanecerá igual.

Jugar en grupo



En primer lugar, se expondrá la **lista** de todas las partidas en grupo del juego en las que usted participa, bien porque haya sido retado por otro alumno, bien porque precisamente es el creador de la partida. Para cada una de ellas puede jugar una única partida, donde obtendrá una nota que competirá con las de sus contrincantes. La forma de juego es exactamente la misma que para el caso individual.

Para crear una **nueva partida en grupo** ha de pulsar en 'Crear nueva partida en grupo' y se mostrará un panel frontal

superpuesto a la página con un formulario a rellenar solicitando la información relativa a la partida. Los datos a cumplimentar son:

- **Nombre de la partida en grupo**, a elegir.
- **Conjunto de usuarios** que son retados, a seleccionar entre la lista de todos los alumnos matriculados en el curso.

Una vez todos los miembros de un grupo han completado su partida, se decidirá el ganador.

Proponer preguntas

Esta sección reúne todas las propuestas de preguntas que haya realizado al profesor en este curso y le permite añadir otras nuevas. Para cada una de ellas se asocia un estado:

- o **Pendiente**, si la propuesta está pendiente de ser aprobada por el profesor.
- o **Aprobado**, si la propuesta ha sido aprobada por el profesor.
- o **Rechazado**, si la propuesta ha sido rechazada por el profesor.

Para aquellas propuestas en estado Pendiente es posible editar su contenido o incluso acceder a su borrado.

Estado	
PENDIENTE	eliminar editar
APROBADO	
APROBADO	
APROBADO	
RECHAZADO	

Ilustración 26: Ejemplo de gestión de preguntas propuestas personales

Y para crear una **nueva propuesta** simplemente haga clic en 'Nueva propuesta de pregunta' y complete el formulario que aparecerá, cuyos campos a rellenar son:



Nueva pregunta ✕

Nombre

Tipo

Contenido de la pregunta

Respuestas
 Añadir respuesta

Tiempo para la pregunta

Guardar Cancelar

Ilustración 27: Panel de Nueva pregunta

- **Nombre** de la pregunta, que será su identificador en las listas de preguntas.
- **Tipo** de la pregunta, que puede ser:
 - **Genérica:** Se trata de una pregunta de tipo test habitual. Respuestas y pregunta son en sí totalmente personalizables.
 - **Verdadero o falso:** Se asociarán automáticamente las palabras “Verdadero” o “Falso” como las respuestas a la pregunta.
 - **Es verdadero:** Se indica que la pregunta en sí toma la forma: “¿Cuál de las siguientes afirmaciones es verdadera?”.
 - **Es falso:** Se indica que la pregunta en sí toma la forma: “¿Cuál de las siguientes afirmaciones es falsa?”.
- **Contenido de la pregunta:** donde se introduce el texto que conforma la pregunta en sí.

- **Respuestas:** permite añadir tantas respuestas como se desee a la pregunta indicando para cada una de ellas, si se trata de una respuesta correcta o incorrecta.



El panel 'Nueva respuesta' tiene un título 'Nueva respuesta' con un botón de cerrar (X) en la esquina superior derecha. Dentro del panel, hay una sección 'Correcta' con un botón de selección (checkbox) a su izquierda. Debajo de esto, hay un campo de texto etiquetado 'Contenido de la respuesta'. En la parte inferior del panel, hay dos botones: 'Guardar' y 'Cancelar'.

Ilustración 28: Panel de Nueva respuesta

- **Tiempo** para la pregunta: se corresponde con el tiempo de que dispondrá el alumno para contestar dicha cuestión. Una cuenta regresiva se encargará de limitar este tiempo durante el juego.

Ver estadísticas personales

En la sección de estadísticas podrá visualizar un análisis de sus resultados obtenidos en el juego. De primeras, se observa una serie de parámetros generales sobre el usuario así como la lista de todas sus participaciones en el juego, junto con algunos datos relativos a ellas. Si desea contemplar más información sobre cada partida pulse sobre 'detalles'.

PARTE 3:

FINALIZACIÓN



Capítulo 7. Presupuesto económico.

Los **costes materiales** de este tipo de trabajos pueden considerarse prácticamente nulos, siempre y cuando, claro está, se realicen desde casa sin contar con gastos de oficina, luz, calefacción, etc. Basta, así, con un ordenador que integre el software adecuado. La **mano de obra**, sin embargo, engloba el tiempo dedicado y los conocimientos adecuados que hacen que no cualquier persona pueda desarrollar el proyecto. Y el precio dado a esto es algo ciertamente **subjetivo**. El tiempo en horas es un factor substancial y bastante objetivo, pero no el único a considerar.

Para proponer un presupuesto económico acorde a los del mercado se atienden a una serie de **claves** que dan una buena idea del precio que tomaría e-Liza v2 [27]:

- En base a las **necesidades del proyecto**, se estima un total de horas dedicadas que dependerá normalmente de la forma de trabajar. De este modo, importa tanto el número de páginas que componen la web como su contenido.
El módulo desarrollado reúne únicamente cinco apartados pero cada uno de ellos posee una cantidad significativa de funcionalidades.
- Según aumenta la **experiencia del desarrollador**, una aplicación web inspira más confianza y, en principio, amplía la garantía.
Mi experiencia personal no permite presumir de grandes hazañas ni demostrar un largo periodo de tiempo de trabajo en esta materia. Pero sí me avalan varias experiencias en el mundo web con buenos resultados, algunas de ellas profesionales.
- El **tipo de cliente** también influye a la hora de presupuestar la aplicación. Grandes clientes podrán permitirse precios más holgados que aquellos más humildes.



Para la Universidad de Valladolid, aun siendo una gran institución, teniendo en cuenta sus fines docentes, sería lo más honesto reducir el coste al máximo posible.

- Atendiendo a la **urgencia** del cliente, también pueden diferenciarse los proyectos entre aquellos que exijan un ritmo normal de trabajo y aquellos que obliguen al apresuramiento. El desarrollo de e-Liza v2, como Trabajo de Fin de Grado que constituye, ha sido posible realizarse, temporalmente hablando, a antojo del desarrollador, aunque con más intensidad en la última etapa.
- Y por último, según la **calidad de vida** del país en que se reside, las tarifas del mercado pueden oscilar mucho, naturalmente. No es que se valore más o menos el trabajo extranjero, pero los diferentes sueldos y gastos repercuten consecuentemente en los precios. En España, posiblemente, el precio del desarrollo web sea bastante más elevado que en los países sudamericanos y en torno a la media de otros países Europeos.

Por otro lado, se lleva a cabo el análisis considerando características más concretas y las funcionalidades particulares que incluye la herramienta web. Por ejemplo, el módulo desarrollado cuenta con una **sección de administración** cuyo acceso queda limitado a los diferentes usuarios del sistema en función de su rol ejercido. En primer lugar, esto debe garantizar un nivel de seguridad significativo que conlleva una cierta responsabilidad. Y en segundo lugar, supone el diseño de una estructura de la base de datos que implica una construcción sólida de la base de la aplicación. Un sistema de administración como el de e-Liza v2, donde residen relativamente complejas relaciones entre tablas y sus delicadas consultas asociadas, puede costar un precio de unos 400€ aproximadamente. A mayores de esta sección, la herramienta consta de **cinco apartados más** con algo menos de complejidad. Pero, puestos a pagar por el conjunto de todas las funcionalidades que abarcan cada uno de ellos, las cuales extienden generosamente las posibilidades del programa, habría que hablar de un precio que



rondase los 200€ de media para cada sección. Los minuciosos detalles de las estadísticas e ideas como el juego en grupo implican un gasto de tiempo a considerar [28].

Por detrás de lo que el cliente ve, existe una lógica de funcionamiento que no es tan trivial como se exterioriza en la interfaz. Y por delante, se presenta un **diseño estético** que también tiene un coste. La imagen de la aplicación es muchas veces tan importante como algunos detalles técnicos. Además, conseguir un producto final que otorgue agilidad a su uso y dé sensación de **profesionalidad y robustez**, también se paga. Así, e-Liza v2 en estos aspectos promete ser bastante competitivo por lo que habría que añadir una cantidad de unos 150 € al presupuesto total.

En suma, el módulo recibe un valor orientativo de unos de 1350€ en mano de obra. Como siempre, esto son cifras imprecisas e inexactas que pueden ascender o disminuir en base a distintos factores. Pero da una idea del presupuesto económico de e-Liza v2.



Capítulo 8. Conclusiones

1. Entre las preocupaciones de las últimas reformas educativas, destacan las que se refieren al cambio metodológico que contribuya a potenciar la mejora y la eficiencia en el proceso de enseñanza-aprendizaje, y las que faciliten la autoevaluación de los alumnos.

En este sentido, el recurso a los "serious game", unido al creciente interés de los alumnos por las nuevas tecnologías, han influido positivamente por su motivación y proactividad en el proceso de enseñanza-aprendizaje, y, por tanto, en el interés por demostrar los conocimientos adquiridos en el aula.

2. En la actualidad existen infinidad de LMS y cada uno de ellos con sus herramientas y funcionalidades. No es posible encontrarlas todas en una única plataforma para, en cada momento, escoger lo que mejor se adapte a cada necesidad. Por ello es muy importante contar con una herramienta de referencia en la que se tenga la confianza de que va a responder a los requerimientos exigidos para satisfacer las necesidades y expectativas presentes y futuras. Moodle ha demostrado ser la más adecuada. Precisamente por eso se ha constituido como una de las plataformas más utilizadas.
3. Se ha conseguido dar satisfacción prácticamente al 100% de los objetivos propuestos al inicio del trabajo con la creación de una nueva herramienta Web multifuncional que servirá de apoyo tanto a la investigación realizada por el Grupo de Telemática e Imagen, como a la docencia, facilitando, entre otras cosas, la autoevaluación de los conocimientos del alumno.
4. Se ha logrado solventar las carencias de las versiones anteriores del mismo módulo, satisfaciendo las exigencias técnicas, estéticas y de seguridad requeridas. De este modo, entre las características destacadas de la nueva versión, señalamos, por ejemplo:



- No existen formas de sortear las reglas del juego que permitan saber la respuesta correcta antes de ser contestada ni reiniciar el contador que limita el tiempo de contestación de una pregunta.
 - La interfaz se presenta como un entorno ágil y cómodo para el usuario.
 - El contenido de las preguntas puede incluir códigos de programación como HTML y no ser interpretados.
 - Cada pregunta puede asociarse a varias etiquetas y puede contener tantas respuestas como se desee.
 - Todas las preguntas de un curso son reutilizables para las distintas instancias de e-Liza v2 del mismo curso.
5. Para llegar a ello, ha sido necesario poner en práctica el conjunto de competencias asociadas al Grado de Ingeniería de telecomunicación. Entre otras:
- Emprendimiento e innovación, desde el momento en que se tomó la decisión de acometer este Proyecto;
 - Aprendizaje autónomo, desarrollando la capacidad de razonamiento crítico que ha permitido detectar las deficiencias en el propio proceso formativo y solventarlas mediante la auto imposición de las actuaciones necesarias para ello.
 - Eficiencia en la comunicación oral y escrita, imprescindible para poder comunicar adecuadamente los resultados del aprendizaje, con capacitación lingüística en inglés para consultar y comprender la bibliografía más actual sobre el objeto del estudio.
 - Capacidad de organización, planificación y gestión del tiempo, necesarios para poder llevar a buen término semejante Proyecto.
6. El resultado del trabajo realizado ha generado una nueva versión del módulo e-Liza que denominamos e-Liza v2 y que aprovecha los últimos avances de las tecnologías web.



7. E-Liza v2 nace con la madurez suficiente como para soportar la prueba de un escenario real de aplicación. Pero la única forma de conocer la calidad y ventajas del sistema es ponerlo en práctica.

Y desde un punto de vista **personal**, es sugerente comentar lo que supone un reto de este calibre para la formación académica exigida para un ingeniero. Un proyecto de esta envergadura trae consigo numerosas **complicaciones** durante su desarrollo. Requiere, en primer lugar, una organización adecuada con un plan de trabajo medido. Esto da muestra de una debida **preparación** y **disposición**. Por otro lado, la propia búsqueda de soluciones ante los posibles baches encontrados, fuerza el progreso del **trabajo autónomo** e **independiente**.

Desde una experiencia tan cercana como la propia, destaca, durante la elaboración del trabajo, la libertad referida a la toma de **decisiones**. Cada paso adelante exige unos argumentos que demuestren que la propuesta elegida, de entre todas las posibles direcciones que podría tomar el desarrollo, es la más adecuada. Así, el hecho de seleccionar las ideas a defender y dirigir el trabajo por un camino y no otro, conlleva una gran **responsabilidad**. No solamente hay que dar la talla manifestando los conocimientos y habilidades relativos a la programación, sino que también hay que parir y dar forma a unas ideas que construyan el mejor producto final posible.

Y para terminar, el resultado más interesante obtenido de la elaboración de este trabajo, es el **continuo aprendizaje** que ha supuesto su desarrollo. Esto quiere decir que cada pequeño avance ha permitido entender mejor el uso de las nuevas tecnologías empleadas y descubrir las soluciones más adecuadas que dan respuesta a cada requisito. De este modo, es posible que algunas de las medidas tomadas durante los comienzos de la elaboración del proyecto, no sean las mismas que las tomadas durante su finalización. Posiblemente algunas decisiones iniciales tienen diferente resolución tras esta experiencia, lo que ha hecho replantear algunos aspectos del trabajo a medida que se ha ido desarrollando, reconstruyendo las partes necesarias.



Capítulo 9. Líneas de futuro

E-Liza v2 se proyecta como herramienta fiable, útil y atractiva para profesores y alumnos. Para los primeros, porque facilita la docencia y porque el trabajo inicial que requiere su aplicación nunca va a ser trabajo perdido, sino que siempre será aprovechable. Y, para los alumnos, porque el método de aprendizaje que se propone encaja perfectamente con la dinámica de entretenimiento de la juventud.

El futuro próximo de e-Liza v2 es su incorporación, de manera progresiva, en las aulas promoviendo de este modo el deseado cambio metodológico.

Por eso es necesario mantener vivo el sistema, ya que los principales consumidores del módulo son también los principales clientes de productos tecnológicos que están en continua actualización.

Para que siga siendo útil a mediano y largo plazo, el desarrollo de e-Liza v2 ha sido llevado a cabo de modo que permita contemplar posibles futuras ampliaciones funcionales o mejoras técnicas. Por ello, se ha seguido unas pautas de programación adecuadas para que el código y la estructura del módulo sean fácilmente comprensibles.

Aunque e-Liza v2 ya incorpora suficientes funcionalidades, siempre es posible su mejora o ampliación, pudiendo plantearse, incluso, la posibilidad de ofrecer futuros Trabajos de Fin de Grado referidos, por ejemplo, a:

- Añadir nuevos parámetros de juego. Como el decremento del valor de las preguntas con el tiempo o segundas oportunidades para volver a intentar dar la respuesta correcta.
- Extender las posibilidades del juego en grupo. Como permitir la competición entre equipos de alumnos o incluso organizar torneos entre ellos.
- Ampliar las funcionalidades de la administración del sistema. Como implementar nuevos sistemas de ordenación de las preguntas: por orden alfabético, por dificultad, por fecha de creación, por etiquetas, etc.



- Incorporar nuevas estadísticas interesantes. Como el uso global que se da al módulo en toda la plataforma por distintos profesores y en un determinado periodo de tiempo; o la presentación de una mayor variedad de gráficas y con más detalle.

En definitiva, un módulo joven y completo que puede aspirar a mantener su juventud durante mucho tiempo.



BIBLIOGRAFÍA

- [1] e-ABC. Empresa dedicada a las soluciones e-Learning. Definición de e-Learning. www.e-abclearning.com/definicion-e-learning. Último acceso: Agosto de 2014.
- [2] Ute Ritterfeld, Michael Cody y Peter Vorderer. Serious games: Mechanisms and effects. 2009.
- [3] Osbaldo Turpo y Margarida Romero. Serious Games para el desarrollo de las competencias del siglo XXI. Revista de Educación a Distancia. Número 34 Pág. 10 de 22. <http://www.um.es/ead/red/34>
- [4] Sitio para el diseño de páginas web. www.desarrolloweb.com. Último acceso: Agosto de 2014.
- [5] Alicia Cañellas Mayor. CMS, LMS y LCMS. Definición y diferencias. Especial CMS de la revista de Comunicación y Pedagogía 251-252. Junio de 2011.
- [6] Juan Manuel Alonso Gutiérrez. Proyecto Fin de Carrera, Aplicación del Aprendizaje Basado en Juegos (GBL): Componente Moodle para Autoevaluación de Conocimientos. Septiembre de 2008.
- [7] Página principal de Moodle, <http://moodle.org/>. Actualización del 11 de julio de 2014. Último acceso: Agosto de 2014.
- [8] Espacio Europeo de Educación Superior. Página oficial. www.eees.es/. Último acceso: Agosto de 2014.
- [9] Sara San Martín del Pozo. Proyecto Fin de Carrera, e-Liza: Componente Moodle para la Autoevaluación Interactiva de Conocimientos. Septiembre de 2009.
- [10] Alfonso Ruifernández Pérez. Proyecto Fin de Carrera, Diseño de gráficos, actualización y despliegue de e-liza: componente moodle para la autoevaluación interactiva de conocimientos. Mayo de 2010.



- [11] Catálogo de plataformas LMS de software libre evaluadas por el proyecto europeo Join. <http://www.ossite.org/join/sp/lms/catalog.htm>. Último acceso: Agosto de 2014.
- [12] Javier Uceda Antolín y Senén Barro Ameneiro (directores). *Universities 2010: Evolución de las TIC en el sistema universitario español 2006-2010*. Conferencia de Rectores de las Universidades Españolas (CRUE). 2010.
- [13] Selwyn, N. The use of computer technology in university teaching and learning: a critical perspective. *Journal of Computer Assisted Learning*, 23(2), 83-94. Abril de 2007.
- [14] Yee, N. The Psychology of MMORPGs: Emotional Investment, Motivations, Relationship Formation, and Problematic Usage. In Schroeder, R. y Axelsson, A. (eds.). *Avatars at Work and Play: Collaboration and Interaction in Shared Virtual Environments (187-207)*. London: Springer-Verlag. 2006.
- [15] Documentación oficial de Moodle. Moodle icons. https://docs.moodle.org/dev/Moodle_icons_2.4. Último acceso: Agosto de 2014.
- [16] Jean-Damien Pogolotti. Página oficial de pChart: librería PHP para la representación de gráficas. <http://www.pchart.net/>. Último acceso: Septiembre de 2014.
- [17] Documentación oficial de Moodle. Getting started. Base module folder. [content.https://docs.moodle.org/dev/NEWMODULE_Documentation](https://docs.moodle.org/dev/NEWMODULE_Documentation). Último acceso: Agosto de 2014.
- [18] Documentación oficial de Moodle. Manual de estilo de código. Reglas generales. https://docs.moodle.org/all/es/Manual_de_Estilo_de_C%C3%B3digo. Último acceso: Agosto de 2014.
- [19] Documentación oficial de Moodle. Coding style. https://docs.moodle.org/dev/Coding_style. Último acceso: Agosto de 2014.



[20] Documentación oficial de Moodle. Database. Database structures. <https://docs.moodle.org/dev/Database>. Último acceso: Agosto de 2014.

[21] Documentación oficial de Moodle. Security. <https://docs.moodle.org/dev/Security>. Último acceso: Agosto de 2014.

[22] Miriam Antón Rodríguez y María Ángeles Pérez Juárez. 'Bases de datos relacionales', documentación de la asignatura de Laboratorio de Desarrollo de Sistemas Telemáticos. Universidad de Valladolid. 2014.

[23] Refsnes Data. W3Schools. CSS3. CSS transitions. http://www.w3schools.com/css/css3_transitions.asp. Último acceso: Septiembre de 2014.

[24] Refsnes Data. W3Schools. HTML5. Attribute input: autofocus. http://www.w3schools.com/tags/att_input_autofocus.asp. Último acceso: Septiembre de 2014.

[25] Refsnes Data. W3Schools. HTML5. Attribute input: placeholder. http://www.w3schools.com/Tags/att_input_placeholder.asp. Último acceso: Septiembre de 2014.

[26] Refsnes Data. W3Schools. HTML5. Form input types. http://www.w3schools.com/html/html5_form_input_types.asp. Último acceso: Septiembre de 2014.

[27] Sitio para el diseño de páginas web. Tarifas de realización de una página web. <http://www.desarrolloweb.com/articulos/2246.php>. Último acceso: Septiembre de 2014.

[28] Sitio para el diseño de páginas web. Tarifas y presupuestos para el desarrollo de webs. <http://www.desarrolloweb.com/manuales/71/>. Último acceso: Septiembre de 2014.

[29] Documentación oficial de Moodle. Roles estándar de Moodle. https://docs.moodle.org/all/es/Roles_est%C3%A1ndar. Último acceso: Septiembre de 2014.



- [30] Miriam Antón Rodríguez y María Ángeles Pérez Juárez. 'PHP', documentación de la asignatura de Laboratorio de Desarrollo de Sistemas Telemáticos. Universidad de Valladolid. 2014.
- [31] Miriam Antón Rodríguez y María Ángeles Pérez Juárez. 'AJAX', documentación de la asignatura de Laboratorio de Desarrollo de Sistemas Telemáticos. Universidad de Valladolid. 2014.
- [32] Miriam Antón Rodríguez y María Ángeles Pérez Juárez. 'CSS3', documentación de la asignatura de Laboratorio de Desarrollo de Sistemas Telemáticos. Universidad de Valladolid. 2014.
- [33] Miriam Antón Rodríguez y María Ángeles Pérez Juárez. 'JavaScript', documentación de la asignatura de Laboratorio de Desarrollo de Sistemas Telemáticos. Universidad de Valladolid. 2014.
- [34] Miriam Antón Rodríguez y María Ángeles Pérez Juárez. 'SQL', documentación de la asignatura de Laboratorio de Desarrollo de Sistemas Telemáticos. Universidad de Valladolid. 2014.
- [35] W3Techs-World WideWeb Technology Surveys. Web especialista en encuestas sobre las tecnologías web. <http://w3techs.com/>. Último acceso: Septiembre de 2014.

ANEXOS



Anexo 1. Roles estándar de Moodle

Un rol es una colección de permisos definida para todo el sistema que pueden ser asignados a usuarios específicos en contextos específicos. Por defecto, Moodle 2.x trae consigo los siguientes roles [29]:

- **Administradores del sitio:** tienen permiso para hacer cualquier acción en el sistema. Al creador del sitio web se le llama administrador primario y no puede ser eliminado de dicho rol, aunque es posible editar sus datos.
- **Mánager (Gestor):** este rol existe desde la versión 2.0 de Moodle. Comparte muchas capacidades del administrador, pero no todas, pudiendo, así, realizar algunas tareas a nivel administrativo.
- **Creador de curso:** puede crear cursos y ejercer de profesor en ellos
- **Profesor (maestro):** puede gestionar y añadir contenidos a los cursos (incluyendo actividades y calificaciones). Además, tiene la capacidad de asignar a otros usuarios el Rol de profesor no-editor o el Rol de estudiante.
- **Profesor no-editor (maestro sin permiso de edición):** puede calificar el trabajo de los alumnos dentro de los cursos, pero no puede gestionarlos (ni alterar ni eliminar ninguna de las actividades o recursos).
- **Estudiante:** puede acceder y participar en actividades del curso y ver recursos, pero no puede gestionar el curso ni ver más que sus propias calificaciones.
- **Invitado:** puede ver cursos que lo permitan, pero no participar en ellos.
- **Usuario autenticado:** rol adicional al resto que adquieren todos los usuarios que han ingresado en el sitio. Se les permite editar su propio perfil, mandar mensajes, escribir en blog y hacer otras cosas además de cursos.
- **Usuario autenticado en la portada:** rol del usuario que ha ingresado al sitio, pero solamente para la portada del sitio.



Anexo 2. PHP

PHP (PHP Hypertext Preprocesor) [30] es un lenguaje de programación interpretado, cuyos comandos se ejecutan en el **servidor** y permiten la creación de documentos HTML dinámicos. Su sintaxis es similar a la de otros lenguajes como C, Perl, Java o JavaScript.

PHP es un producto de **código abierto**, lo que quiere decir que se puede acceder a su código, usarlo, modificarlo y distribuirlo de forma gratuita sin que suponga coste alguno, al contrario de lo que ocurre con los productos comerciales. Esta es, sin duda, una de las principales ventajas de PHP, ya que la comunidad de usuarios que desarrollan mejoras para el sistema es enorme, lo que proporciona a PHP una vitalidad y capacidad de adaptación que no poseen otros lenguajes de programación. Toda la evolución de PHP puede seguirse en el sitio web www.php.net.

PHP se encuentra disponible para muchas plataformas incluyendo Windows, Unix o Linux y con la ventaja de que las aplicaciones desarrolladas en PHP se pueden transportar de una plataforma a otra sin necesidad de modificaciones, es decir que PHP presenta una portabilidad elevada. A pesar de ser en la actualidad un lenguaje **multiplataforma**, debe comentarse que su entorno nativo es Unix/Linux. Con respecto al servidor web también existirían diferentes opciones, ya que podría usarse por ejemplo IIS (Internet Information Server) de Microsoft o Apache, este último con la ventaja de que, al igual que PHP, es también un producto de libre distribución.

Entre las ventajas de PHP se encuentra también la de permitir la **conexión** y el acceso a multitud de **sistemas gestores de bases de datos** como MySQL, PostgreSQL u Oracle.

PHP fue concebido en 1994 por Rasmus Lerdorf con la intención de crear un contador para averiguar el número de visitas que recibía su CV virtual. Sin embargo, con el tiempo ha sido adoptado por otros desarrolladores que lo han transformado y convertido en la herramienta que es hoy en la actualidad. Una herramienta utilizada ya en más de

200 millones de dominios de Internet, número que **sigue creciendo** y cuya evolución responde a la siguiente gráfica:

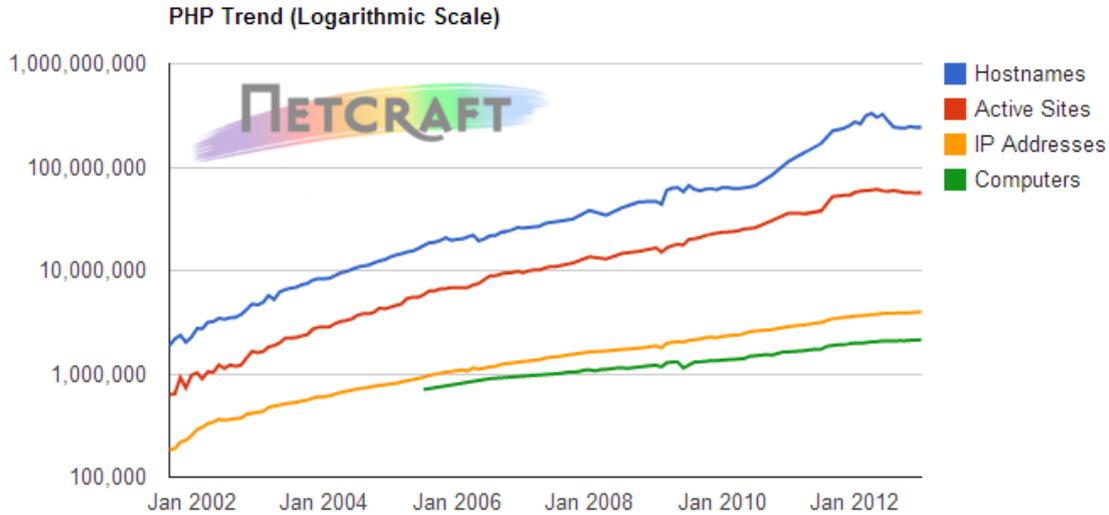


Ilustración 29: Evolución del uso de PHP

Concretamente se ha disparado el uso de la última versión de PHP como puede verse en la siguiente gráfica:

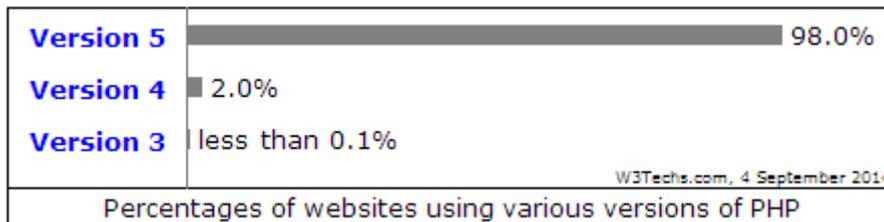


Ilustración 30: Uso de PHP según versiones

PHP es utilizado por el 81.2% (septiembre, 2014) de los sitios web de los que se conoce el lenguaje del lado del servidor utilizado. En la siguiente figura se puede ver la gráfica comparativa con otros de los principales lenguajes de programación de lado del servidor [35].

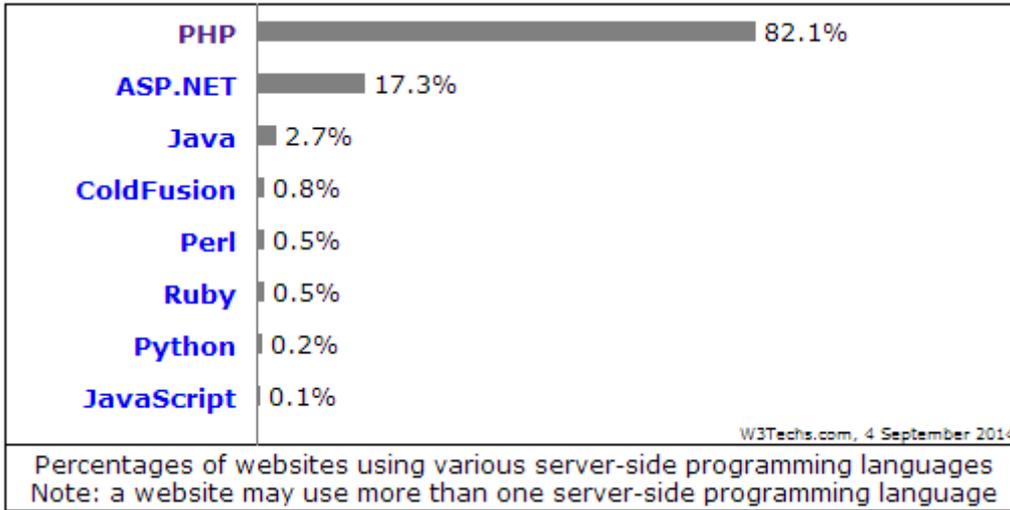


Ilustración 31: Gráfica comparativa de diferentes lenguajes de programación del lado del servidor

En la ilustración inferior se puede observar el uso dado a cada lenguaje de programación del lado del servidor considerando el número de sitios web desarrollados en cada lenguaje y su tráfico [35].

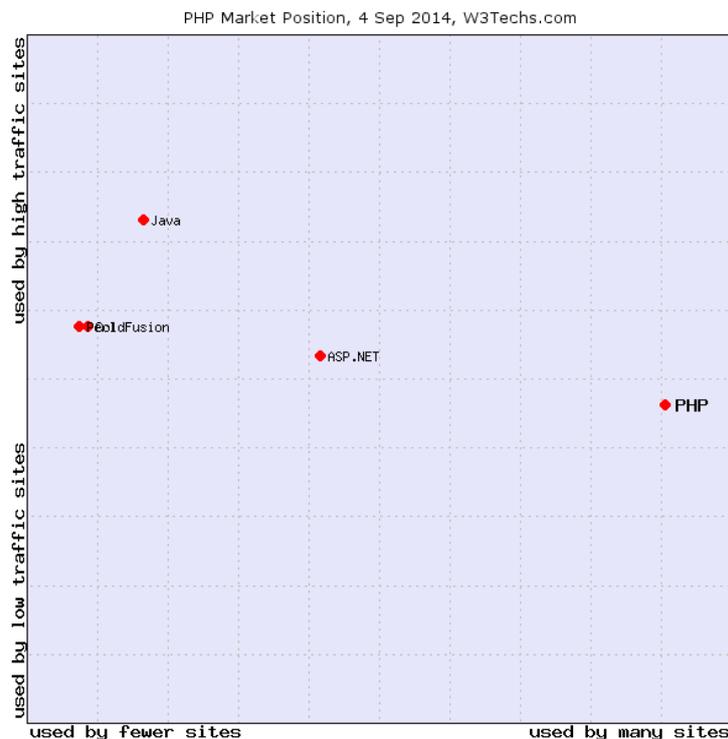


Ilustración 32: Relación de la extensión de distintos lenguajes de programación con su tráfico

Una tecnología que aparezca cerca de la esquina inferior derecha se utiliza en muchos sitios web aunque con un tráfico medio. Una tecnología situada cerca de la esquina superior izquierda se utiliza en pocos sitios web, pero principalmente con un tráfico alto. Las mejores posiciones se encuentran cerca de la esquina superior derecha. En concreto, PHP se utiliza en muchos sitios web del tráfico medio.

Algunos **ejemplos** de sitios web que emplean PHP son los siguientes:

- phpMyAdmin
- SquirrelMail
- OScommerce
- Magento
- eCommerce
- Moodle
- Drupal
- Joomla
- Gallery
- MediaWiki
- Facebook
- Tuenti
- WordPress
- Elgg
- Digg
- Sourceforge
- MyYearbook

Por último, comentar el aspecto funcional del lenguaje. El código PHP se puede insertar en un documento HTML, siendo así el proceso de **generación** de una página **web dinámica** con PHP el que se describe a continuación:

- Cuando un usuario hace clic desde su navegador en un enlace correspondiente a un documento HTML que contiene código PHP, el navegador realiza la solicitud al servidor web correspondiente.
- El servidor web localiza entonces el documento, detecta que contiene código PHP y pone en funcionamiento el intérprete del lenguaje.
- Dicho intérprete ejecuta el código PHP y genera un resultado, generalmente en forma de página web, que se devuelve al navegador para que éste se encargue de su visualización.

Anexo 3. AJAX

AJAX (Asynchronous JavaScript y XML) [31] **no es** un nuevo **lenguaje** de programación, sino una nueva forma de utilizar herramientas ya existentes. Su principal objetivo y ventaja es el de permitir el intercambio de pequeñas cantidades de datos entre cliente y servidor con el fin de realizar la **actualización de forma asíncrona de partes de una web** sin tener que recargarla de forma completa. De esta forma, AJAX permite realizar páginas web más **dinámicas** proporcionando una experiencia ágil de navegación al usuario.

Fue popularizado por Google en el 2005 gracias a la aplicación Google Suggest que usaba esta tecnología para crear una interfaz web dinámica de forma que si se comenzaba a teclear en el campo de búsqueda de Google, se enviaba un conjunto de letras al servidor, el cual devolvía una lista de sugerencias. Actualmente existen numerosas aplicaciones que utilizan AJAX.

La siguiente figura muestra el funcionamiento básico de AJAX. Así, cuando un evento tiene lugar, (1) se crea un **objeto XMLHttpRequest** y se envía una **petición HttpRequest** al **servidor**, el cual procesa dicha petición, (2) crea una **respuesta** y devuelve los datos oportunos al navegador que, a su vez, (3) procesa los datos devueltos mediante el uso de **JavaScript** y actualiza el contenido de la página.

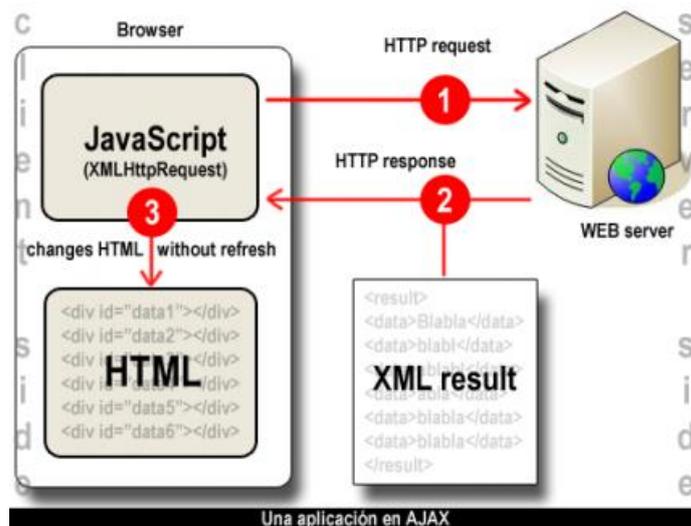


Ilustración 33: Proceso de funcionamiento de AJAX

Figura: Esquema de funcionamiento de AJAX.

Las principales herramientas usadas por AJAX son:

- El objeto **XMLHttpRequest**, que permite intercambiar datos de forma asíncrona con el servidor
- **JavaScript** y **DOM** (Document Object Model o Modelo de Objetos del Documento) para interactuar y mostrar la información.
- **CSS** (Cascading Style Sheets u Hojas de Estilo en Cascada) para especificar el estilo de los datos actualizados que van a mostrarse en el navegador.
- **XML** (Extensible Markup Language o Lenguaje de Marcado Extensible) que es el formato con frecuencia utilizado para transferir los datos. Aunque e-Liza v2 utiliza transferencias de los resultados de procesamientos de ficheros **PHP**.

Anexo 4. CSS3

Las hojas de estilo facilitan el **control de la presentación** o estilo de las páginas web, y su posterior revisión y actualización en caso de ser necesarias, **separando** para ello la información relativa a la **presentación** de la relativa a la **estructura**. Además, las hojas de estilo representan un avance importante para los autores de páginas web, al darles un mayor rango de posibilidades para controlar la **aparición** de sus páginas [32].

En los entornos científicos en los que la Web fue concebida, existía una mayor preocupación por la estructura de los documentos que por su presentación. A medida que la Web era descubierta por un espectro más amplio de usuarios de distintas procedencias, se vio la necesidad de superar las limitaciones estilísticas de HTML. Aunque las intenciones eran buenas – mejorar la presentación de las páginas web –, algunas técnicas para conseguirlo tenían efectos secundarios negativos. Entre estas técnicas, que dan buenos resultados en algunas circunstancias, pero no siempre ni en todos los casos, se incluyen:

- Conversión del texto en imágenes, cuando el texto incorpora tipos de fuentes poco habituales.
- Utilización de imágenes para controlar el espacio en blanco.
- Utilización de tablas para la organización de las páginas.
- [...]

Las hojas de estilo resuelven este problema al tiempo que superan el limitado rango de mecanismos de presentación de HTML. Con las hojas de estilo es más fácil, por ejemplo, especificar la cantidad de espacio entre líneas, los colores a utilizar para el texto y el fondo, el tamaño y tipo de las fuentes y otros muchos detalles relativos al estilo o presentación. Se puede afirmar que HTML informará al navegador de cómo se estructura el documento mientras que CSS le informará de cómo debe renderizarlo.

En principio, se podría usar cualquier sintaxis para la información de estilo en los documentos cuya estructura se especifique mediante HTML. Los autores pueden especificar la sintaxis de la información de estilo

utilizada mediante el elemento <META> que permite especificar la sintaxis usada por defecto para la información de estilo incorporada en el documento. Por ejemplo, para especificar que el lenguaje de estilo por defecto es CSS (Hojas de Estilo en Cascada, Cascading Style Sheets), los autores deberían especificar en la cabecera del documento lo que aparece a continuación:

```
<META HTTP-EQUIV="content-style-type" CONTENT="text/css">
```

Los clientes deberían determinar el **lenguaje de estilo por defecto** siguiendo los siguientes pasos:

- Si algún elemento <META> especifica content-style-type como valor del atributo HTTP-EQUIV, el último de ellos en el flujo del documento determina el lenguaje de estilo por defecto.
- De otro modo, el lenguaje de estilo por defecto es "text/css".

Si se emplean hojas de estilo en cascada, existen **tres formas de añadir información de estilo** a las páginas web:

- Añadiendo instrucciones de estilo a elementos concretos, mediante el atributo STYLE.
- Incluyendo las instrucciones de estilo en la cabecera de un documento HTML concreto, mediante el elemento <STYLE>. Esto permite controlar la apariencia de todo el documento HTML mediante un conjunto de instrucciones de estilo.
- Enlazando un documento HTML, o varios, o todos los documentos HTML que componen un sitio web, mediante el elemento <LINK>, a una hoja de estilo externa que contenga la definición del estilo. De esta forma, se puede modificar la apariencia de una, múltiples o todas las páginas web de un sitio web, revisando un solo fichero.

En el diseño de un sitio web se pueden utilizar uno o más métodos simultáneamente.

Dejando ahora de un lado los aspectos más técnicos, la antigua versión de este lenguaje, **CSS2.1**, reunía muchas características que permitían lograr bien los objetivos marcados. Pero pronto surgió **CSS3**, última versión a día de hoy, para extender el conjunto de elementos estéticos

disponibles, dotando al navegador web de la capacidad de controlar casi cualquier aspecto de presentación.

La implementación CSS3 se ha realizado de forma que esta versión es completamente compatible con las versiones anteriores de CSS, por lo que no resulta necesario modificar estilos ya diseñados. Esto supone que las diferentes versiones de navegadores que soporten CSS3, soportarán también CSS2.1.

Por el contrario, debe tenerse en cuenta que no todas las versiones de navegadores usadas actualmente implementan de forma completa los estilos propuestos por CSS3. Además, a la hora de aplicar los nuevos estilos propuestos por CSS3, es importante tener en cuenta que buena parte de ellos aún se encuentran en fase de desarrollo, por lo que no hay establecido un estándar para su uso. Esto quiere decir que cada navegador los aplica a veces de manera diferente, incluso con algunos matices en los parámetros que emplean. Para poder utilizar estos nuevos estilos en periodo de experimentación es necesario emplear lo que se conoce como *vendor prefixes*, que son prefijos que se añaden delante de la propiedad CSS que se vaya a aplicar. Existe uno para cada navegador, aunque algunos de ellos coinciden debido a que diferentes navegadores utilizan el mismo motor de renderizado. Los principales vendor prefixes son los siguientes:

- **Android:** `-webkit-`.
- **Chrome:** `-webkit-`.
- **Firefox:** `-moz-`.
- **Internet Explorer:** `-ms-`.
- **iOS:** `-webkit-`.
- **Opera:** `-o-`.
- **Safari:** `-webkit-`.

Ejemplos de propiedades CSS3 en fase de experimentación para las que es necesario utilizar los vendor prefixes son los que se muestran a continuación:

- `-webkit-column-count:3;`
- `-moz-box-sizing: border-box;`

CSS3 se divide en módulos que incluyen las características de la versión anterior CSS2.1, así como otras características nuevas propuestas en CSS3. Algunos de los principales módulos propuestos son los siguientes:

- Fondos y Bordes – Backgrounds and Borders –
- Efectos de Texto – Text Effects –
- Transformaciones 2D / 3D – 2D/3D Transformations –
- Animaciones – Animations –
- Diseño con Múltiples Columnas – Multiple Column Layout –
- Interfaz de Usuario – User Interface –

Por último, se presentan, muy brevemente, algunos de los más destacados **ejemplos** concretos de las **nuevas características** que ofrece CSS3:

- Define estilos que controlan la **redondez de las esquinas** de muchos elementos de HTML y otros tipos de complicadas transformaciones en los bordes.
- Introduce la capacidad de agregar **sombras paralelas** tanto a contenedores como segmentos de texto HTML.



Ilustración 34: CSS3. Propiedades: border-radius y shadows

Ejemplo: Drop shadows + border-radius

- Introduce poderosos métodos para controlar la **opacidad** y el **color** de contenido HTML añadiendo espacios de color con soporte para HSL (Hue, Saturation, Lightness), HSLA (HSL con Alpha), RGB (Red, Green, Blue) y RGBA (RGB con Alpha).
- Añade el soporte de **media queries**, que permiten a diseñadores y desarrolladores crear estilos que se adapten a dimensiones específicas de pantalla.



Ilustración 35: CSS3. Media queries

- Añade nuevos **selectors**: reglas condicionales para aplicar hojas de estilo a los elementos HTML.
- Inclusión de **transformations**: estilos que controlan la transformación visual de un objeto en particular (2D y 3D).
- Implementa **animación** y **transiciones**, que dotan de movimiento a los elementos y de cambios de estado visuales.

Anexo 5. JavaScript

JavaScript [33] es un **lenguaje de script**. Un script, o guión realizado en JavaScript puede insertarse en un documento HTML dotando a la página web de cierto grado de **interactividad**, ya que puede proporcionar, por ejemplo, un mecanismo para la detección y tratamiento de eventos como un clic con el dispositivo apuntador – normalmente el ratón – o el deslizamiento del dispositivo apuntador sobre una determinada zona.

Al igual que HTML, JavaScript es un lenguaje **interpretado por el cliente** y, por tanto, no existe restricción alguna en el servidor. Cada evento a ser procesado no requiere comunicación con el servidor lo que permite, por ejemplo, realizar un filtrado de la información recogida mediante un formulario antes de enviarla a una base de datos para ser almacenada.

JavaScript tampoco depende de otras condiciones de contorno como el Sistema Operativo (S.O.) pues, como ya se ha comentado, es un lenguaje que interpreta el cliente, con lo que los scripts funcionarán en todo S.O. que cuente con un cliente preparado para ello.

Los bloques de código JavaScript pueden **insertarse** en los documentos **HTML** de varias formas:

- Mediante el uso de la etiqueta <SCRIPT> de HTML
- Mediante el empleo de manejadores de eventos
- Mediante el empleo del URL JavaScript

Un guión JavaScript se compone de **sentencias**, instrucciones o proposiciones que son órdenes que se dan al intérprete del lenguaje y que finalizan con un punto y coma.

En cuanto a la **popularidad** del lenguaje, basta con observar el porcentaje de páginas web que usan cada uno de los principales lenguajes de programación del lado del cliente, donde JavaScript aplasta a tus competidores con el 88.3% de uso a día de hoy (septiembre, 2014) [35].

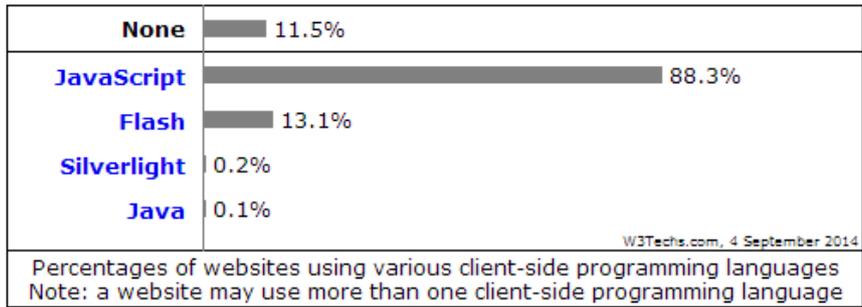


Ilustración 36. Uso de los principales lenguajes de programación del lado del cliente

Anexo 6. MySQL y SQL

MySQL [22] es un sistema gestor de bases de datos relacional altamente extendido y es usado por algunos de los sitios web más grandes y populares del mundo: Wikipedia, Google, Facebook, Twitter, Flickr y Youtube.

Un **Sistema Gestor de Bases de Datos (SGBD)** es un programa (o conjunto de programas) que, básicamente, permite a los usuarios acceder a la Base de Datos para almacenar información en la misma o para realizar operaciones sobre la información contenida en ella relacionadas con su recuperación, modificación o eliminación.

No hay que confundir el concepto **Base de Datos** con el de Sistema Gestor de Bases de Datos. Así, si se ve la Base de Datos como un archivador de datos, el Sistema Gestor de Bases de Datos debería verse como el conjunto de programas mediante el cual se manipula y gestiona dicho archivador.

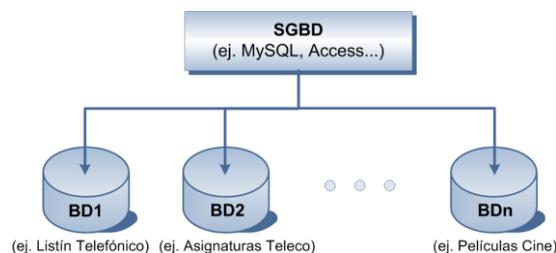


Ilustración 37. Relación Bases de datos y Gestores de Bases de Datos

Para el ejemplo que nos ocupa, e-Liza v2, podemos considerar que la base de datos es el conjunto de todas las listas de preguntas, respuestas, etiquetas, etc. del juego. Mientras que el sistema gestor de la base de datos utilizado en el trabajo es, precisamente, MySQL. Aunque bien podrían haberse utilizado otros ejemplos como Postgre SQL u Oracle, admitidos por XMLDB, la capa de abstracción de la base de datos que utiliza Moodle.

MySQL utiliza, para su gestión, el lenguaje **SQL (Structured Query Language)** [34]. Se trata de un lenguaje de programación diseñado precisamente para almacenar, manipular y recuperar datos



almacenados en bases de datos relaciones. SQL apareció por primera vez en 1974 cuando Codd (IBM) desarrolló el primer prototipo de una base de datos relacional.

Existe un SQL estándar, pero los diferentes sistemas gestores de bases de datos lo implementan con ligeras diferencias entre sí. Esto se debe a dos motivos:

- El estándar es bastante complejo y no es práctico implementarlo de forma completa.
- Cada proveedor de sistemas gestores de bases de datos quiere diferenciar su producto del resto.

Este lenguaje de consulta tiene dos partes fundamentales:

- **DML (Data Manipulation Language)** usado para obtener, borrar, actualizar, añadir, etc., datos en la base de datos.
- **DDL (Data Definition Language)** usado para creación/destrucción de tablas, etc., en la base de datos.



Anexo 7. Código de generación de los datos de la gráfica temporal.

Este código calcula los datos necesarios para después representar, mediante la librería pChart, la **nota media** del conjunto total de alumnos del curso **agrupada por meses** en el rango de fechas en que aquellos han participado del juego.

El vector *\$data* será el encargado de almacenar los valores de la gráfica y el vector *\$labels* almacenará las etiquetas del eje de abscisas, correspondientes a los meses.

```
//Consulta de todas las sesiones de la instancia del e-Liza
$sessions = $DB->get_record_sql('SELECT MAX(closingdate) as maxdate,
MIN(closingdate) as mindate FROM mdl_eliza_sessions WHERE
elizaid='.$elizaaid.'");
//Obtención de datos temporales
$minyear = date('Y', $sessions->mindate);
$minmonth = date('n', $sessions->mindate);
$maxyear = date('Y', $sessions->maxdate);
$maxmonth = date('n', $sessions->maxdate);
//vector de datos
$data = array();
//vector de etiquetas
$labels = array();
for ($j=$minyear; $j<=$maxyear; $j++) {
    if ($minyear==$maxyear) {
        for ($i=$minmonth; $i<=$maxmonth; $i++) {
            $limit1 = strtotime("$j-$i-01");
            $limit2 = strtotime("$j-".($i+1)."-01");
            $session = $DB->get_record_sql('SELECT AVG(score) as
avgscore FROM mdl_eliza_sessions WHERE elizaid='.$elizaaid.' AND
closingdate>='.$limit1.'" AND closingdate<'.$limit2);
            if ($session->avgscore) {
                $data[] = $session->avgscore;
            } else {
                $data[] = VOID;
            }
            $labels[] = "$i/$j";
        }
    } else {
        if ($j==$minyear) {
            for ($i=$minmonth; $i<=12; $i++) {
                $limit1 = strtotime("$j-$i-01");
                $limit2 = strtotime("$j-".($i+1)."-01");
                $session = $DB->get_record_sql('SELECT AVG(score) as
avgscore FROM mdl_eliza_sessions WHERE elizaid='.$elizaaid.' AND
closingdate>='.$limit1.'" AND closingdate<'.$limit2);
                if ($session->avgscore==0) {
```


Anexo 8. Código de generación de la lista de usuarios matriculados en el curso.

Para obtener la lista de usuarios matriculados en un curso primeramente hay que entender una parte importante de la base de datos. El foro oficial de Moodle ha ayudado a reclutar la siguiente información acerca de las **tablas** que intervienen en la consulta:

- **mdl_role**: aquí está la relación de los roles existentes. De aquí necesitaremos el id de cada rol para ver qué usuarios hay matriculados para cada curso con cada rol. Si sólo queremos los estudiantes, habría que quedarse sólo con el identificador del rol Estudiante.
- **mdl_course**: en esta tabla se almacenan los cursos. Basta con buscar aquí por el fullname (o el shortname) el curso que deseamos ver sus matriculados, y a través de ahí podemos sacar el identificador del curso. Aunque el código cuenta de inicio ya con este identificador.
- **mdl_context**: aquí están relacionados las diferentes instancias de los contextos que hay. Cada contexto tiene un código numérico (llamado nivel ó contextlevel) que lo identifica. En concreto, para los cursos, el nivel es el 50, por lo que, sabiendo cuál es el identificador del curso, basta con buscar la entrada con el instanceid=idcurso y contextlevel=50, y el identificador que aparezca para esos datos será el identificador del contexto. Sin embargo, también existe la función `context_course::instance(courseid)` que nos permite obtener lo mismo de una forma más directa.
- mdl_role_assignments**: por último esta es la tabla donde están las relaciones de las matriculaciones. Basta con hacer una búsqueda por el contextid deseado (es decir, el identificador del contexto del curso que hemos sacado de mdl_context), y por el identificador del rol que queramos (estudiante, por ejemplo), y nos dará los identificadores de todos los usuarios matriculados.

El código final resultante es el mostrado a continuación, donde tiene especial interés la consulta SQL que tiene lugar en la cuarta línea:



```

//Obtención del contexto del curso
$contextcourse = context_course::instance($cm->course);
//Consulta en la bases de datos de los estudiantes matriculados
$students = $DB->get_records_sql("SELECT a.userid, u.lastname,
u.firstname FROM mdl_role_assignments a INNER JOIN mdl_user u ON
a.userid = u.id WHERE a.contextid = ".$contextcourse->id." AND a.roleid
IN (SELECT id FROM mdl_role WHERE archetype = 'student') AND a.userid
<>".$USER->id);
//Generación de la tabla
$response=='';
foreach ($students as $student) {
    if ($response=='') {
        $response="<table>
            <tr>
                <td>Nombre</td>
                <td>Retar</td>
            </tr>";
    }
    $response = $response . "
        <tr>
            <td>".$student->lastname.", ".$student->
>firstname."</td>
            <td><a onclick='challengeUser(\".$student->
>id.') '>Retar</a></td>
        </tr>";
    } //end foreach
if ($response!='') {
    $response=$response."</table>";
} else {
    $response="<div>No existe ningún usuario matriculado en el
curso</div>";
}
echo $response;

```

Anexo 9. Diagrama de casos de uso

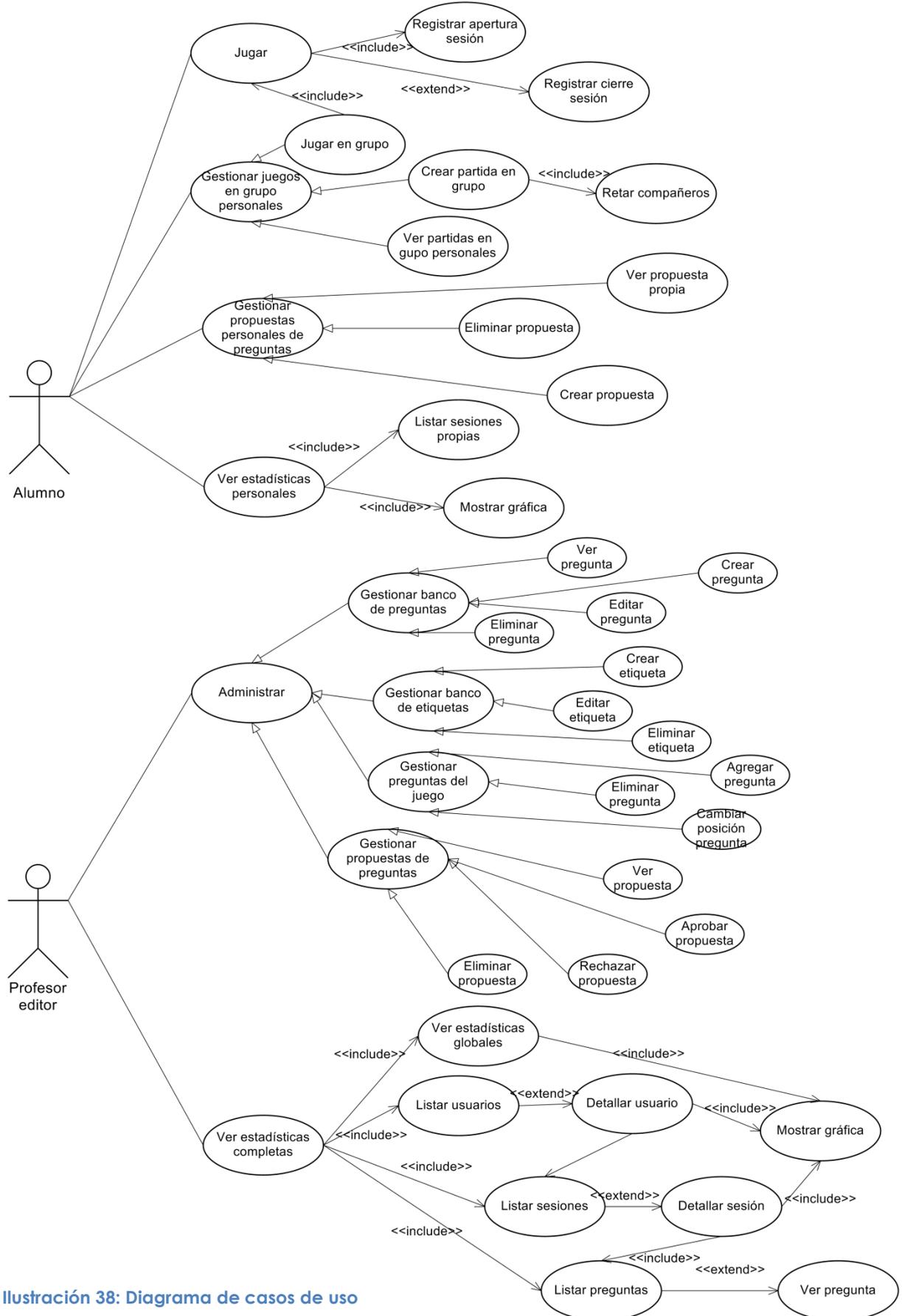


Ilustración 38: Diagrama de casos de uso

Anexo 10. Diagramas de actividades UML

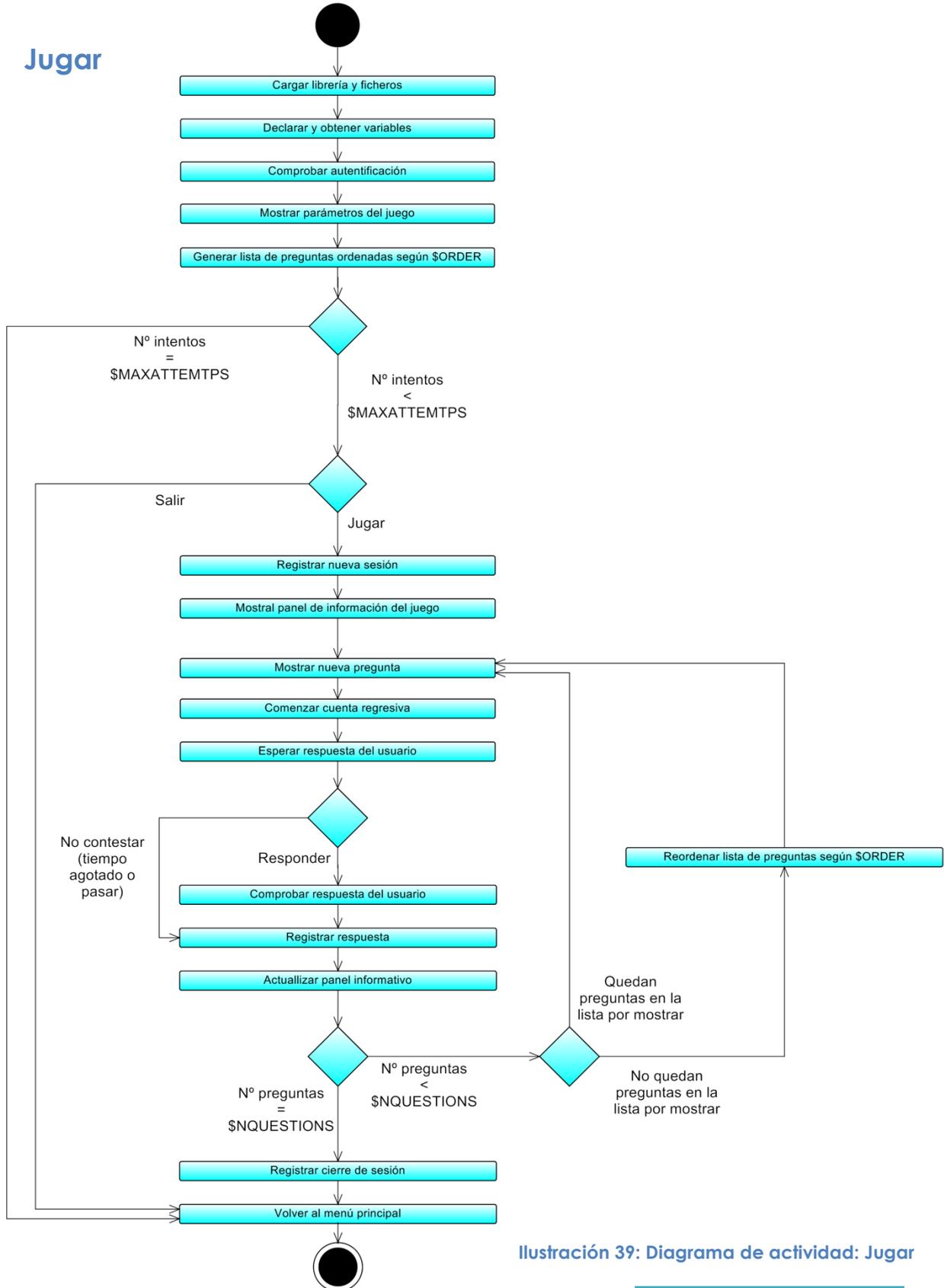


Ilustración 39: Diagrama de actividad: Jugar

Jugar en grupo

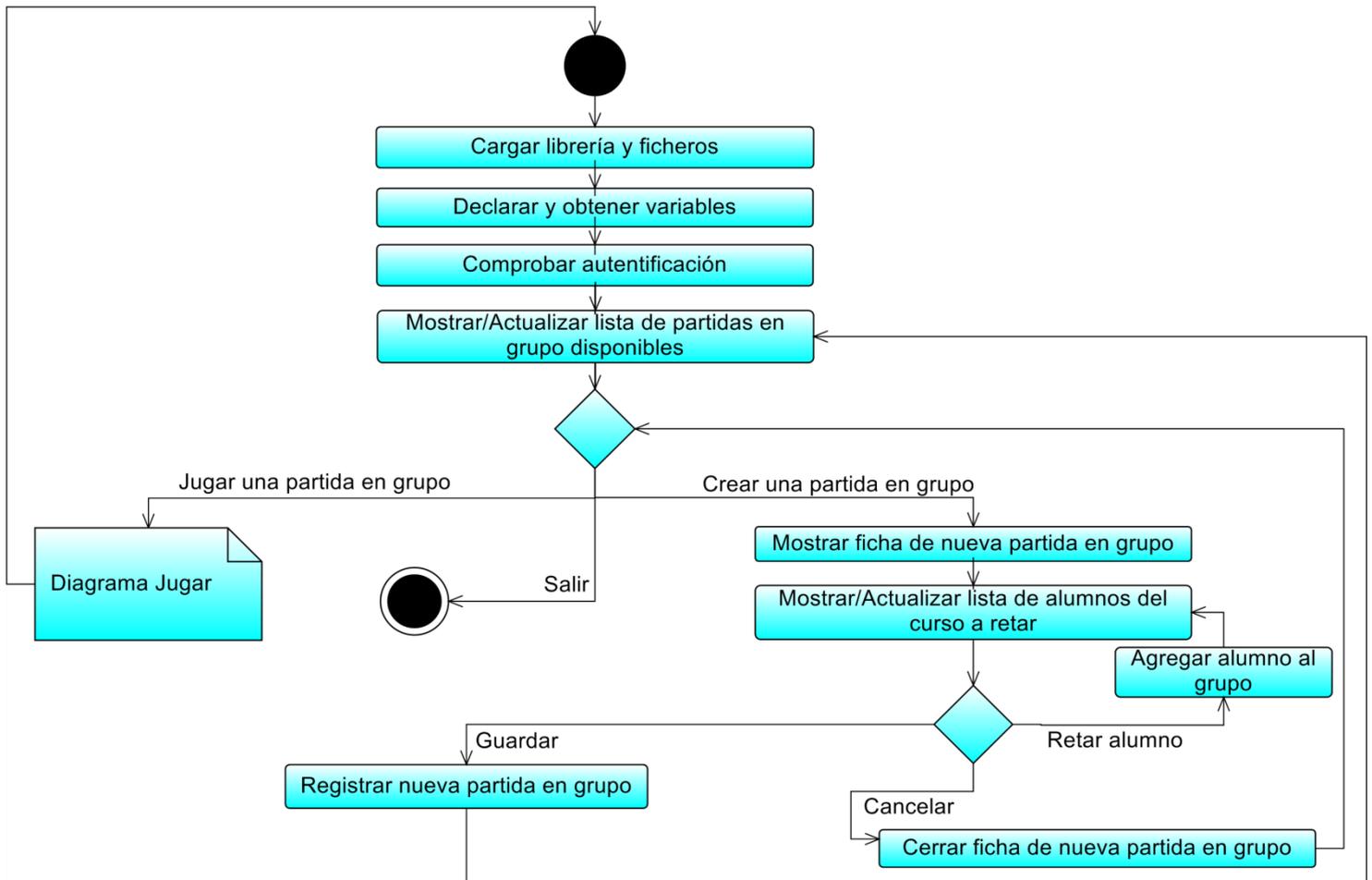


Ilustración 40: Diagrama de actividad: Jugar en grupo

Administrar

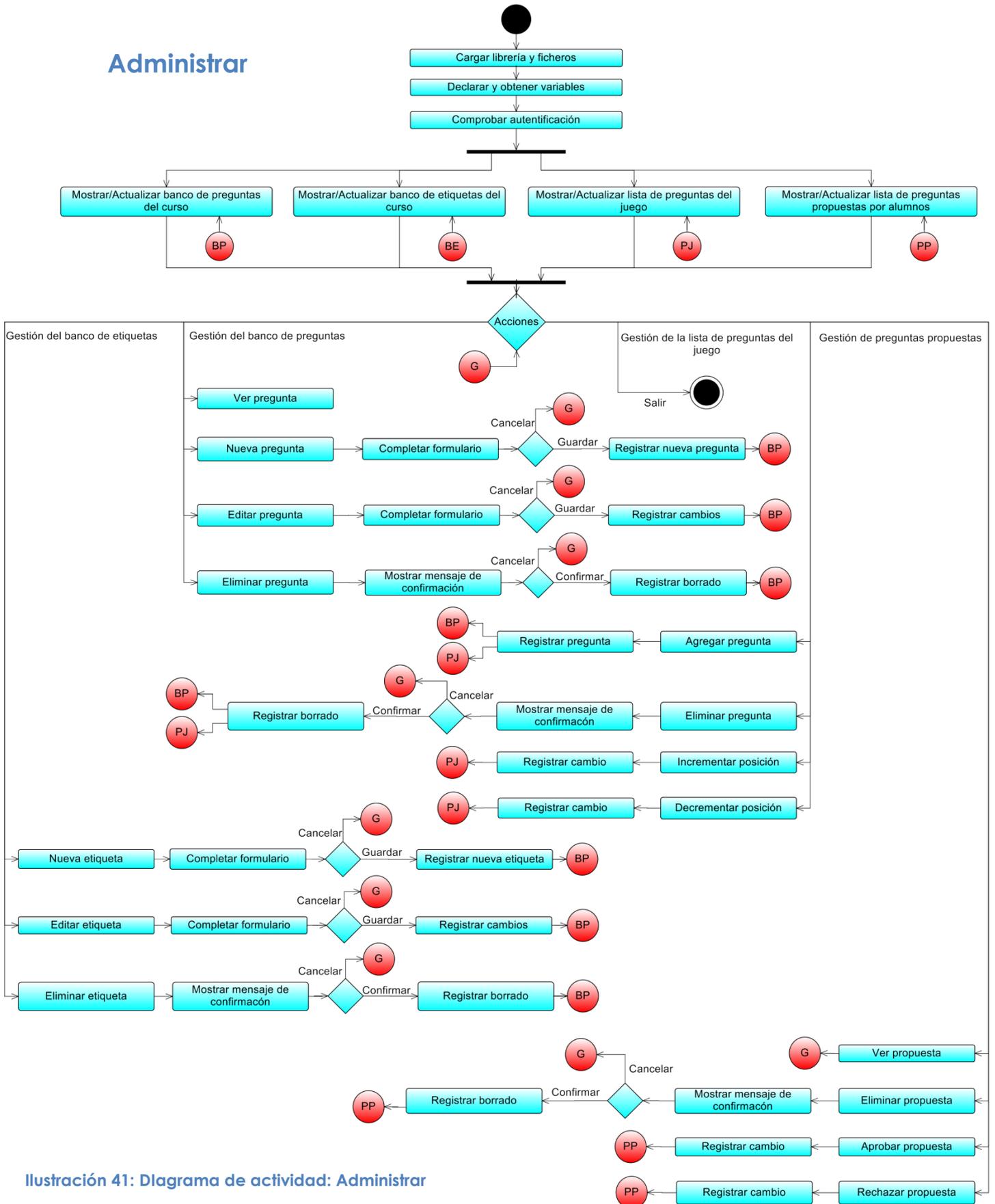


Ilustración 41: Diagrama de actividad: Administrar

Proponer preguntas

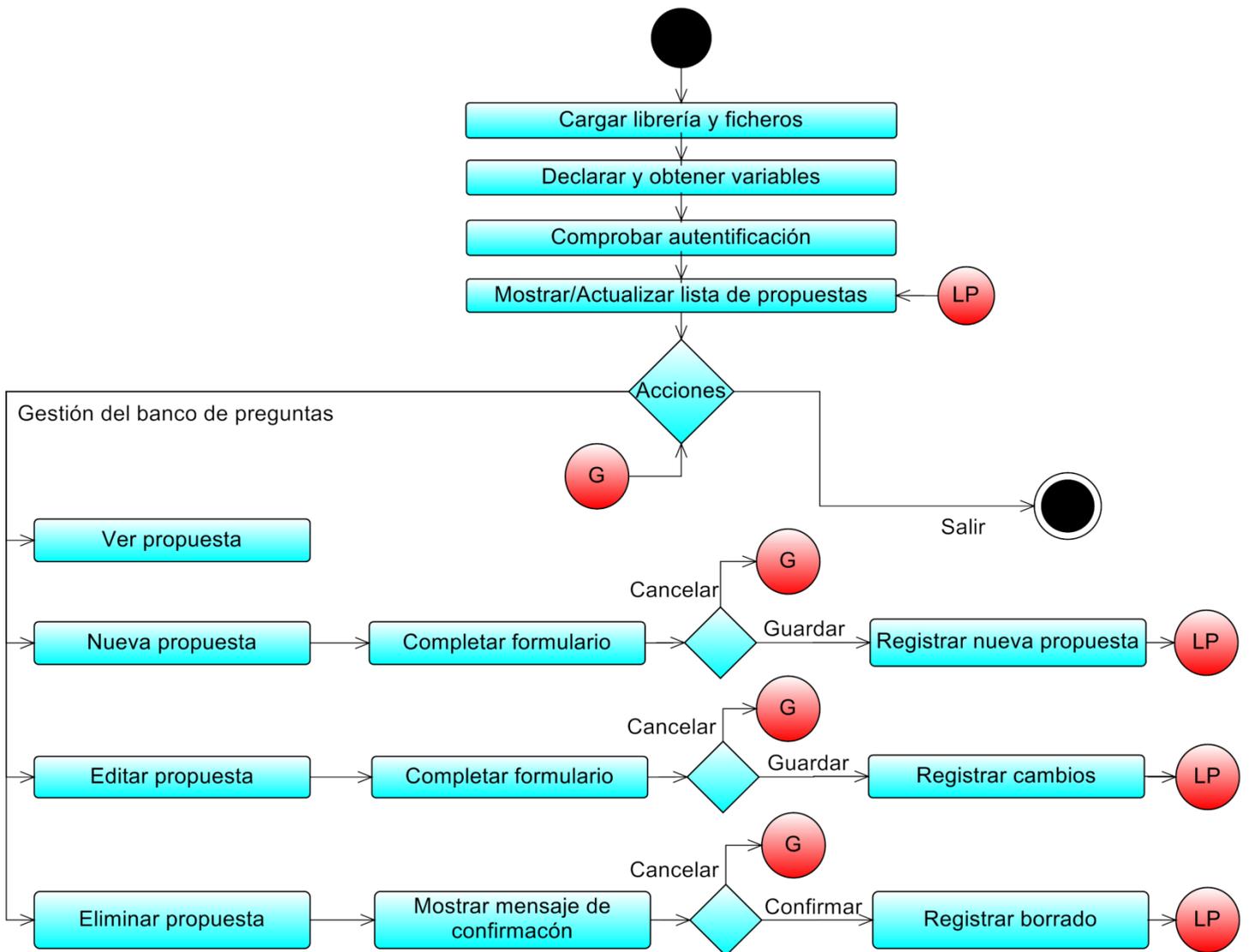


Ilustración 42: Diagrama de actividad: Proponer preguntas

Estadísticas

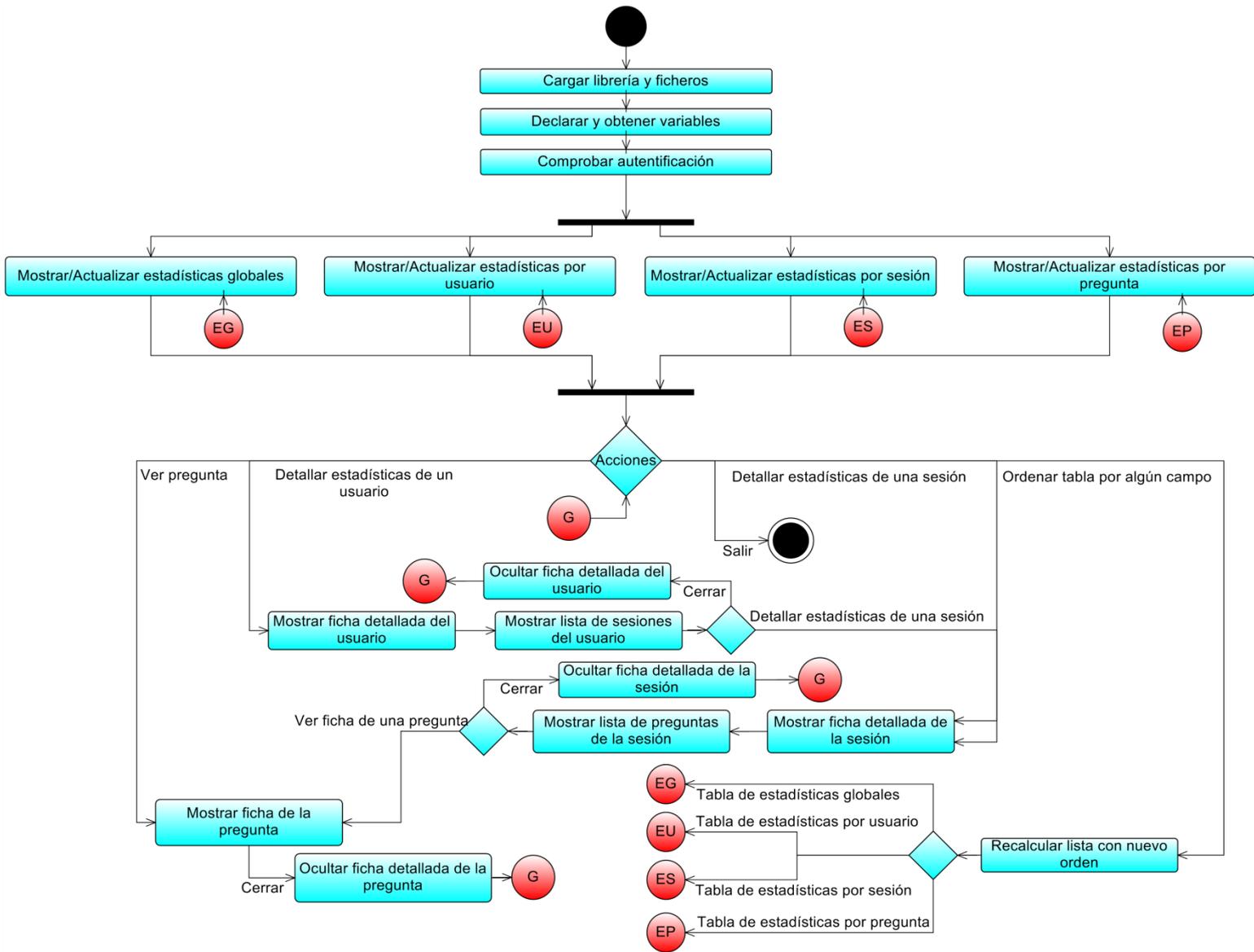


Ilustración 43: Diagrama de actividad: Estadísticas

