

UNIVERSIDAD DE



VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

## TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE  
TELECOMUNICACIÓN, MENCIÓN EN TELEMÁTICA

# **Creación de un módulo de actividad docente georreferenciada para Moodle**

Autor:

**D. Adrián Rodríguez Fernández**

Tutor:

**D. Juan Pablo de Castro Fernández**

Valladolid, 18 de julio de 2016

---

TÍTULO: **Creación de un módulo de actividad docente georreferenciada para Moodle**  
AUTOR: **D. Adrián Rodríguez Fernández**  
TUTOR: **D. Juan Pablo de Castro Fernández**  
DEPARTAMENTO: **Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

---

**TRIBUNAL**

---

PRESIDENTE: **D. Juan Pablo de Castro Fernández**  
VOCAL: **Dña. María Jesús Verdú Pérez**  
SECRETARIO: **Dña. Luisa María Regueras Santos**  
SUPLENTE: **D. Francisco Merino Caminero**  
SUPLENTE: **D. Alonso Alonso Alonso**

---

FECHA: **Julio de 2016**  
CALIFICACIÓN:

---

## **Resumen de TFG**

En este Trabajo de Fin de Grado se pretende desarrollar una herramienta e-Learning que sirva de soporte a la comunidad docente para la organización de actividades georreferenciadas.

La idea nace fruto de la necesidad de responder a la evolución de los métodos docentes, promovidas por el rápido avance de las tecnologías y los nuevos planes de estudio. En este contexto, el juego de la caza del tesoro ofrece la posibilidad de combinar la enseñanza fuera de las restricciones del aula, al mismo tiempo que amplía la motivación y el entusiasmo tanto de profesores como de alumnos. Así, particularizando el trabajo, se plantea el desarrollo de un nuevo módulo para la plataforma educativa Moodle que adapte al plano tecnológico el juego de la caza del tesoro mediante el uso de las tecnologías móviles, aportando soluciones y ventajas con respecto a la creación y desarrollo del juego en su formato tradicional.

Para ello, se hace uso de un amplio abanico de tecnologías y librerías que facilitan el desarrollo web y la gestión de los datos espaciales.

### **Palabras clave**

E-learning, Moodle, Caza del tesoro, Ajax, PHP, JavaScript, JQuery, CSS3, SQL, Openlayers 3.

### **Abstract**

In this Final Project it is pretended to develop an e-Learning tool that would serve to support the teaching community for the organization of georeferenced activities.

The idea borns as a result of the necessity to respond to the evolution of the teaching methods, promoted by the quick progress of technologies and the new curricula. In this context, the treasure hunt game offers the possibility of combining the education out of the restrictions of the classroom, at the same time that it broadens the motivation and enthusiasm of both teachers and pupils. Thus, specifying the work, it is raised the development of a new module for the educational platform Moodle that would adapt to the technological field the treasure hunt game by using mobile technologies, providing solutions and advantages with regard to the creation and development of the game in its traditional format.

For this purpose, a wide range of technologies and libraries that make easier the web development and the management of spatial data are used.

### **Keywords**

E-learning, Moodle, Treasure hunt, Ajax, PHP, JavaScript, JQuery, CSS3, SQL, Openlayers 3.

*A mi familia, especialmente a mis padres y mi hermana, por todo el apoyo y cariño recibido durante los años de carrera.*

*A Rebeca, por saber aguantarme en los momentos de estrés y aportarme la motivación y serenidad que necesitaba.*

*Por último y no menos importante a Juan Pablo de Castro por darme esta oportunidad*

# Índice de contenidos

Índice de contenidos .....	I
Índice de figuras .....	IV
<b>1. Introducción .....</b>	<b>1</b>
1.1. Motivación.....	1
1.2. Objetivos.....	1
1.3. Fases y métodos.....	2
1.4. Estructura de la memoria .....	3
<b>2. Estado del arte .....</b>	<b>5</b>
2.1. E-learning .....	5
2.2. Moodle.....	7
2.3. Juego de la caza del tesoro .....	8
2.4. Aplicaciones similares existentes .....	8
2.4.1. <i>Treasure-HIT</i> .....	8
2.4.2. <i>GeoCaching</i> .....	10
2.4.3. <i>7Scenes</i> .....	11
2.4.4. <i>Skattjat</i> .....	12
2.4.5. <i>fAR-Play</i> .....	12
<b>3. Análisis del problema .....</b>	<b>14</b>
3.1. Introducción.....	14
3.2. Requisitos del trabajo .....	14
3.2.1. <i>Requisitos globales del módulo</i> .....	15
3.2.2. <i>Requisitos de configuración del juego</i> .....	15
<b>4. Desarrollo del proyecto .....</b>	<b>17</b>
4.1. Aspectos técnicos .....	17
4.1.1. <i>Configuración del entorno de trabajo</i> .....	17
4.1.2. <i>Directrices del código</i> .....	18
4.1.2.1. <i>Reglas Generales</i> .....	18
4.1.2.2. <i>Estilo del código</i> .....	19
4.1.2.3. <i>Estructura de la base de datos</i> .....	20
4.1.2.4. <i>Normas de seguridad</i> .....	21
4.1.3. <i>Estructura de ficheros del módulo</i> .....	22
4.1.4. <i>Diseño del modelo de datos</i> .....	34
4.1.4.1. <i>Tabla treasurehunt</i> .....	38
4.1.4.2. <i>Tabla treasurehunt_locks</i> .....	39
4.1.4.3. <i>Tabla treasurehunt_roads</i> .....	40

4.1.4.4.	<i>Tabla treasurehunt_stages</i> .....	40
4.1.4.5.	<i>Tabla treasurehunt_answers</i> .....	42
4.1.4.6.	<i>Tabla treasurehunt_attempts</i> .....	42
4.2.	Aspectos funcionales .....	44
4.2.1.	<i>Administración de la actividad</i> .....	44
4.2.1.1.	<i>Configuración general</i> .....	44
4.2.1.2.	<i>Entorno de creación y edición</i> .....	47
4.2.2.	<i>Jugar</i> .....	59
4.2.3.	<i>Otras funcionalidades</i> .....	65
<b>5.</b>	<b>Manual de usuario</b> .....	<b>66</b>
5.1.	Manual para el administrador .....	66
5.1.1.	<i>Instalación del módulo</i> .....	66
5.1.1.1.	<i>Instalación directamente desde el directorio de plugins de Moodle.</i> .....	66
5.1.1.2.	<i>Instalación mediante un archivo ZIP subido al sitio</i> .....	67
5.1.1.3.	<i>Instalación manual en el servidor</i> .....	67
5.1.2.	<i>Desinstalación del módulo</i> .....	67
5.2.	Manual para el profesor .....	68
5.2.1.	<i>Creación de una actividad</i> .....	68
5.2.2.	<i>Pantalla de configuración de la actividad</i> .....	69
5.2.3.	<i>Pantalla de inicio</i> .....	71
5.2.3.1.	<i>Información general de la caza del tesoro</i> .....	71
5.2.3.2.	<i>Progreso de los usuarios</i> .....	71
5.2.4.	<i>Pantalla de edición de la caza del tesoro</i> .....	74
5.2.4.1.	<i>Funcionalidades de la interfaz de edición</i> .....	74
5.2.4.2.	<i>Pantalla de edición de caminos</i> .....	78
5.2.4.3.	<i>Pantalla de edición de etapas</i> .....	79
5.3.	Manual para el alumno .....	80
5.3.1.	<i>Pantalla de inicio</i> .....	80
5.3.1.1.	<i>Información general de la caza del tesoro</i> .....	80
5.3.1.2.	<i>Historial de intentos del usuario/grupo</i> .....	80
5.3.1.3.	<i>Progreso de los usuarios</i> .....	80
5.3.2.	<i>Pantalla de juego</i> .....	81
5.3.2.1.	<i>Funcionalidades de la interfaz del juego</i> .....	81
5.3.2.2.	<i>Primera etapa</i> .....	86
5.3.2.3.	<i>Modo revisión del juego</i> .....	87
5.3.2.4.	<i>Notificaciones</i> .....	87
<b>6.</b>	<b>Conclusiones y líneas futuras</b> .....	<b>90</b>
6.1.	Conclusiones .....	90
6.2.	Líneas futuras .....	91
<b>7.</b>	<b>Referencias</b> .....	<b>93</b>
<b>8.</b>	<b>Anexos</b> .....	<b>96</b>

8.1.	PHP.....	96
8.2.	JavaScript .....	96
8.3.	AJAX.....	97
8.4.	CSS.....	98

# Índice de figuras

Figura 2.1	Página de configuración de una estación en Treasure-HIT .....	9
Figura 2.2	Interfaz de usuario de Treasure-HIT .....	9
Figura 2.3	Contenedor de un geocache típico .....	10
Figura 2.4	Interfaz de usuario de Geocaching .....	11
Figura 2.5	Interfaz de usuario de 7Scenes .....	11
Figura 2.6	Interfaz de usuario de Skattjat .....	12
Figura 2.7	Ejemplo de imagen mostrada con realidad aumentada en fAR-Play .....	13
Figura 4.1	Estructura de ficheros del módulo la caza del tesoro. ....	23
Figura 4.2	Formato de una cadena de ejemplo en español. ....	27
Figura 4.3	Formato de una cadena de ejemplo en inglés. ....	27
Figura 4.4	Icono principal del módulo. ....	28
Figura 4.5	Contenido del fichero versión.php .....	29
Figura 4.6	Esquema de la base de datos. ....	37
Figura 4.7	Petición en formato JSON al servidor web del módulo .....	49
Figura 4.8	Respuesta en formato JSON al servidor web del módulo .....	50
Figura 4.9	Diagrama de flujo de la funcionalidad de edición .....	58
Figura 4.10	Diagrama de flujo de la funcionalidad de jugar .....	64
Figura 5.1	Versión de Moodle en la página de notificaciones .....	66
Figura 5.2	Instalación manual en el servidor .....	67
Figura 5.3	Desinstalación del módulo .....	68
Figura 5.4	Creación de la actividad .....	68
Figura 5.5	Pantalla de inicio al entrar por primera vez como profesor. ....	71
Figura 5.6	Aviso de camino no validado .....	72
Figura 5.7	Aviso de camino sin usuarios asignados .....	72
Figura 5.8	Aviso de camino sin grupos asignados .....	72
Figura 5.9	Aviso de usuarios sin ningún camino asignado .....	72
Figura 5.10	Aviso de alumnos con más de un camino asignado .....	73
Figura 5.11	Aviso de grupos con más de un camino asignado .....	73
Figura 5.12	Selección del historial de intentos de un usuario/grupo. ....	73
Figura 5.13	Interfaz de edición de la caza del tesoro .....	74
Figura 5.14	Buscador de localización en la interfaz de edición. ....	75
Figura 5.15	Cambio de relación de tamaño entre el contenedor de etapas y el mapa ....	76
Figura 5.16	Cuadro de diálogo con información de la etapa. ....	76
Figura 5.17	Etapas de un camino bloqueado. ....	76
Figura 5.18	Interfaz de edición con un camino que tiene menos de 2 etapas. ....	77
Figura 5.19	Interfaz de edición con una etapa sin geometría. ....	78
Figura 5.20	Interfaz de juego de la caza del tesoro .....	81
Figura 5.21	Panel de información con la descripción de la actividad, con una pista bloqueada y con otra sin bloquear. ....	82
Figura 5.22	Panel de búsquedas de localización .....	82
Figura 5.23	Cuadro de diálogo con información sobre una etapa superada. ....	83
Figura 5.24	Cuadro de diálogo con la fecha sobre una localización fallida. ....	83
Figura 5.25	Icono de la última ubicación del usuario. ....	84
Figura 5.26	Pantalla de historial de intentos en el juego .....	84



Figura 5.27 Pantalla de selección de vista del mapa en el juego.....	84
Figura 5.28 Proceso de envío y validación de una ubicación en el juego .....	85
Figura 5.29 Proceso de envío y validación de la respuesta a una pregunta en el juego.	86
Figura 5.30 Proceso para descubrir la primera etapa.....	86
Figura 5.31 Notificación por cambios producidos por los compañeros de equipo. ....	87
Figura 5.32 Notificaciones por cambios en el modo de juego. ....	88
Figura 5.33 Notificación de actividad a completar superada. ....	88
Figura 5.34 Notificación de tiempo superado para realizar la actividad. ....	89
Figura 5.35 Notificación de error. ....	89
Figura 8.1 Funcionamiento de la tecnología AJAX .....	98

# Índice de tablas

Tabla 1: treasurehunt .....	39
Tabla 2: treasurehunt_locks.....	40
Tabla 3: treasurehunt_roads.....	40
Tabla 4: treasurehunt_stages .....	42
Tabla 5: treasurehunt_answers .....	42
Tabla 6: treasurehunt_attempts.....	43

# 1. Introducción

## 1.1. Motivación

Hoy en día, los sistemas educativos están bajo una creciente presión para responder y adaptarse a la rápida innovación tecnológica y a los cambios asociados a la forma en que trabajamos y vivimos. De ello nace una necesidad fundamental para mejorar los procesos de aprendizaje a través de la evolución de los planteamientos pedagógicos, con el fin de hacer que el aprendizaje sea más atractivo y, según se espera, más efectivo.

Una oportunidad de atraer una atención particular son los “Juegos Serios”. Esta modalidad se define como juegos cuyo objetivo principal es la educación, por encima del entretenimiento, y ofrecen un considerable potencial para facilitar tanto el aprendizaje formal como informal. Los juegos serios pueden ofrecer un “puente” entre el sistema educativo y la actual generación de jóvenes estudiantes, que están en riesgo de caer en una brecha cada vez mayor entre los estilos de vida “en red” y el relativo estancamiento que experimentan en la escuela y la universidad.

En este sentido, se ha demostrado que las experiencias similares al juego tienen el poder de motivar a los estudiantes a participar en su aprendizaje y pueden proporcionar una adecuada plataforma para muchas actividades al aire libre. El juego de "La caza del tesoro" es uno de los enfoques que puede combinar la docencia fuera de las restricciones del aula, al mismo tiempo que amplía la motivación y el entusiasmo de los alumnos. Durante el juego, los estudiantes se enfrentan a la resolución de pistas para encontrar la ubicación de los lugares de interés dispuestos por los profesores, hasta alcanzar la ubicación final dónde descubren el “tesoro”. Además, el juego de la caza del tesoro permite jugar de acuerdo a diferentes escenarios: individualmente o en grupos. De esta forma, el matiz del juego puede tomar un carácter competitivo que impulse a los alumnos a involucrarse a la vez que permite aportar valores como la colaboración cuando el modo de juego es en grupos. También proporciona un rico potencial pedagógico que incluye en gran medida el aprendizaje exploratorio y de investigación, sin olvidarnos de los beneficios que aporta como actividad física al aire libre.

Con este contexto y el contacto y conocimientos adquiridos sobre la información georreferenciada durante el desarrollo de la asignatura de Infraestructura de Datos Espaciales (IDE) de la Universidad de Valladolid, se ha llevado a cabo el propósito de convertir el juego de la caza del tesoro al plano tecnológico. Las actuales plataformas de *e-learning* (aprendizaje electrónico), entre las que destaca Moodle en el ámbito universitario, junto con el uso de las últimas tecnologías presentes en los dispositivos móviles proporcionan el marco ideal para su desarrollo y adaptación.

## 1.2. Objetivos

El propósito final de este Trabajo Fin de Grado es el diseño e implementación de un módulo docente para la plataforma Moodle que permita la adaptación del conocido juego “La caza del tesoro” a los dispositivos móviles con capacidad de georreferenciación y a los procesos docentes apoyados en plataformas telemáticas de *e-learning*, ampliando su

funcionalidad y características. De esta manera se espera conseguir un método de aprendizaje más visual y práctico que anime a los estudiantes a explorar su entorno mientras juegan. A su vez, se quiere expandir las herramientas docentes al alcance de los profesores para la realización de actividades al aire libre, y animarles a crear oportunidades de aprendizaje para sus estudiantes más allá de los límites del aula tradicional.

Otros objetivos generales que se esperan conseguir con la realización de este proyecto son:

- ❖ Poner a prueba el conjunto de las competencias alcanzadas durante el estudio del grado.
- ❖ Incrementar las capacidades autodidactas para resolver los problemas que surjan durante el desarrollo del proyecto.
- ❖ Formarse en los últimos avances y tecnologías existentes para la implementación del trabajo.

### **1.3. Fases y métodos**

Las fases que se han seguido para desarrollar este proyecto son:

#### **❖ Proceso de documentación**

En este sentido, se estudia el contexto del proyecto, teniendo en cuenta la plataforma que lo soportará y los conceptos necesarios para su desarrollo en ella. Concretamente, se analiza detalladamente la API (Application Programming Interface) de Moodle y las características básicas y avanzadas que ofrece para la programación de un módulo de actividad.

#### **❖ Planificación de la solución**

En este punto se debe pensar en los requisitos mínimos que debe cumplir cada una de las funcionalidades de la aplicación, así como las tecnologías o librerías que pueden ayudar en su implementación. También se comprueba la viabilidad de cada idea surgida, descartando aquellas que por motivos técnicos no puedan llevarse a cabo.

#### **❖ Puesta en marcha del entorno de trabajo**

Esta fase analiza y selecciona las herramientas de programación necesarias para crear el módulo de actividad para Moodle. En la elección, es prioritario una plataforma de desarrollo que permita la creación de código en los lenguajes de programación soportados por Moodle y los sistemas para realizar pruebas sobre el programa creado.

#### **❖ Desarrollo y pruebas**

Esta etapa abarca simultáneamente tanto el desarrollo en sí del módulo de actividad como su depuración, ya que parece impensable crear funcionalidades sin comprobar a la vez que actúan como cabe esperar. Para realizar las pruebas de la aplicación correctamente se debe utilizar un navegador web compatible con Moodle que actúe como cliente de la

plataforma y realizar un exhaustivo análisis de la funcionalidad del módulo con los diferentes roles para los que está diseñado, obteniendo la retroalimentación necesaria para solucionar errores.

## **1.4. Estructura de la memoria**

Este documento se estructura en siete apartados principales cuyo contenido se detalla a continuación:

### **1. Introducción**

Se introduce el tema abordado en el proyecto describiendo los motivos de su puesta en marcha, su finalidad y la metodología para llevarlo a cabo. Y, al final, se plantea esta estructura de la memoria, con el contenido que se verá en cada apartado.

### **2. Estado del arte**

Se presenta el contexto del proyecto, abordando el concepto de *e-learning* y sus características. También se explican las características de la plataforma impuesta para la construcción del módulo: Moodle, los matices del juego “La caza del tesoro” y las aplicaciones similares a lo que se intenta realizar.

### **3. Análisis del problema**

Se estudian las dificultades actuales para la realización del juego de la caza del tesoro y se marcan unos requisitos de trabajo a conseguir durante la elaboración del proyecto.

### **4. Desarrollo del proyecto**

Se describen los aspectos técnicos que se han tenido en cuenta a la hora de crear la aplicación, tales como la configuración del espacio de trabajo, las pautas y recomendaciones que aporta Moodle para el desarrollo de un módulo, la estructura de ficheros y el esquema de la base de datos que contendrá la caza del tesoro. También se plantean los aspectos funcionales que describen cómo se han implementado cada una de las funcionalidades presentes en el módulo.

### **5. Manual de usuario**

Se introduce al usuario en la actividad mediante una guía que recorre cada una de las pantallas del juego desde la perspectiva de los potenciales roles de la aplicación.

### **6. Conclusiones y líneas futuras**

Se valora el grado de consecución de los objetivos planteados al principio, así como las competencias demostradas para el desarrollo del trabajo. También se mencionan las líneas futuras que puede alcanzar el proyecto, tanto en materia de proyección como en posibles actualizaciones y mejoras.

### **7. Referencias**

En este apartado se incluyen todas las referencias mencionadas a lo largo del documento, tanto para la elaboración de la memoria como para las herramientas que han posibilitado la realización del proyecto.

## **8. Anexos**

Se detallan las principales tecnologías empleadas para el desarrollo del proyecto, como son PHP, JavaScript, AJAX y CSS.

## 2. Estado del arte

### 2.1. E-learning

El *e-learning*, también traducido como aprendizaje electrónico en español, es una modalidad de enseñanza-aprendizaje que consiste en el diseño, puesta en práctica y evaluación de un curso o plan formativo desarrollado a través de redes de ordenadores y puede definirse como una educación o formación ofrecida a individuos que están geográficamente dispersos o separados o que interactúan en tiempos diferidos del docente empleando los recursos informáticos y de telecomunicaciones. Lo característico del *e-learning* es que el proceso formativo tiene lugar totalmente o en parte a través de una especie de aula o entorno virtual en el cual tiene lugar la interacción profesor-alumnos, así como las actividades de los estudiantes con los materiales de aprendizaje [1].

Este nuevo concepto se está expandiendo rápidamente por todo el sistema educativo impregnando tanto la educación formal como la no formal, tanto la enseñanza presencial como la formación a distancia. Por ello, actualmente, la clásica concepción del e-learning como educación a distancia tiene que ser matizada y redefinida ya que el uso de los espacios y aulas virtuales también son empleados en las diversas modalidades de educación presencial.

Las posibilidades que ofrece esta modalidad educativa son diversas, pero se pueden resumir en las siguientes [1]:

- ❖ Extender los estudios y formación a colectivos sociales que por distintos motivos no pueden acceder a las aulas convencionales.
- ❖ Acceder permanentemente a variadas y múltiples fuentes y recursos de información más allá del profesor y del libro de texto.
- ❖ Innovar y cambiar de procesos de aprendizaje por recepción a procesos constructivistas del conocimiento.
- ❖ Flexibilizar el horario escolar y los espacios para el desarrollo de actividades de docencia y aprendizaje.
- ❖ Facilitar la colaboración entre docentes y estudiantes más allá de los límites físicos y académicos del centro educativo al que pertenecen.
- ❖ Abaratar los costes, tanto en la transmisión del material didáctico como la necesidad de personal e infraestructuras necesarias para la docencia.

Una plataforma *e-learning* es el software servidor que proporciona una serie de herramientas que permiten el aprendizaje en línea. Actualmente existen plataformas de *e-learning* en todos los niveles educativos y cuentan con una gran diversidad de funcionalidades, entre las que se encuentra la autenticación al sistema, foros, acceso a contenidos, etc.

Existen varios tipos de plataformas *e-learning* atendiendo al sistema de gestión de contenidos que utilizan. Concretamente se pueden dividir en tres grandes grupos: CMS (Content Management System), LMS (Learning Management System) y LCMS (Learning Content Management System) [2].

- ❖ Los CMS son, dentro de las plataformas de *e-learning*, los más básicos y permiten la generación de sitios web dinámicos. El objetivo de estos programas es la creación y gestión de información en línea (textos, imágenes, gráficos, vídeos, sonido, etc.). También se caracterizan por no poseer herramientas elaboradas de colaboración (foros, chats, diarios, etc.) ni apoyo en tiempo real.
- ❖ Los LMS aparecen a partir de los CMS y proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación de la web con la colaboración de múltiples usuarios. Están orientados al aprendizaje y la educación, proporcionando herramientas para la gestión de contenidos académicos, permitiendo mejorar las competencias de los usuarios de los cursos y su intercomunicación, en un entorno donde es posible adaptar la formación a los requisitos de la empresa y al propio desarrollo profesional. Disponen de herramientas que permiten la distribución de cursos, recursos, noticias y contenidos relacionados con la formación general.
- ❖ Los LCMS son plataformas que integran las funcionalidades de los CMS y los LMS, que incorporan la gestión de contenidos para personalizar los recursos de cada estudiante y donde las empresas se convierten en su propia entidad editora, con autosuficiencia en la publicación del contenido de una forma sencilla, rápida y eficiente, resolviendo los inconvenientes de las anteriores plataformas. Ofrecen facilidad en la generación de los materiales, flexibilidad, adaptabilidad a los cambios, control del aprendizaje y un mantenimiento actualizado del conocimiento.

Además de distinguirse por el tipo de sistema gestor empleado, las plataformas *e-learning* pueden ser clasificadas dependiendo de las licencias de software utilizadas, dando lugar por un lado a las plataformas comerciales, en las cuales es necesario pagar para poder utilizarlas, y por otro a las plataformas *open-source* (código abierto), las cuales pueden ser utilizadas sin costes y suelen ser desarrolladas por entidades educativas.

Las ventajas de este tipo de servicio también pueden ser aprovechadas a través de los teléfonos móviles o tablets gracias al *m-learning*, el cuál aporta una serie de ventajas como puede ser la oportunidad de aprender en cualquier lugar y cualquier momento, un precio más reducido de la tecnología utilizada, etc. A pesar de ello puede presentar una serie de inconvenientes como pueden ser problemas de usabilidad o la gran diversidad existente entre dispositivos.

Evidentemente, para el desarrollo de nuestro trabajo nos centraremos en las plataformas *e-learning* de código abierto que nos permitan crear módulos (o actividades) que posteriormente puedan ser añadidos y reutilizados en los cursos, para lo que necesitaremos un sistema gestor de tipo LCMS. Dentro de este grupo, destaca la plataforma elegida Moodle.



## 2.2. Moodle

Moodle es una plataforma LCMS (Learning Content Management System) enfocada a la docencia. Fue creada y diseñada por Martin Dougiamas para proporcionar a educadores, administradores y estudiantes un sistema integrado único, gratis, robusto y seguro para crear ambientes de aprendizaje personalizados en línea.

La estructura básica de Moodle está organizada en torno a los cursos. Un curso en Moodle es un área en donde un profesor puede añadir recursos y actividades para que sus alumnos las completen. Cada curso puede estar gestionado por varios profesores, que controlan todo su contenido y su duración; pueden tener contenido para un año de estudios, para una sesión única o para cualquier variante. Los estudiantes pueden ser matriculados en el curso manualmente por el profesor, automáticamente por el administrador, o se les puede permitir inscribirse a ellos mismos. Los cursos también permiten unir a los estudiantes en grupos, y a su vez aglomerar grupos en agrupaciones que podrán ser utilizados dentro de las actividades para realizar tareas por equipos o para restringir el acceso a ciertos usuarios.

Existen distintos módulos de actividad disponibles dentro de un curso. Por un lado, están las actividades básicas que forman parte de Moodle y que permiten garantizar una base asentada sobre la que construir todo tipo de conocimientos, como son las tareas, los cuestionarios, las encuestas, los foros, etc. Por otro lado, Moodle proporciona las herramientas necesarias a desarrolladores para construir sus propias actividades y luego enlazarlas con su directorio de *plugins*, con el fin de poder ser descargadas, instaladas y utilizadas por todo el mundo.

El progreso de los usuarios en un curso puede visualizarse de varias formas:

- ❖ Mediante calificaciones: Cada curso tiene su propio libro de calificaciones dónde se almacena la nota de cada estudiante para las actividades calificables.
- ❖ Mediante competencias: Las competencias definen el nivel de comprensión sobre un tema. Cada competencia dentro de un curso puede asignarse a una o más actividades, que mediante su realización proporcionan al profesor una valoración sobre el nivel de destreza alcanzado por cada alumno.
- ❖ Mediante finalizaciones de actividad: El profesor puede especificar los criterios que deben satisfacer las actividades para considerarse como completadas por los alumnos.

La forma de gestionar los permisos de un usuario en la plataforma se realiza a través de roles. Un rol es una colección de permisos definida para todo el sistema que puede ser asignada a usuarios específicos en contextos específicos. La combinación de roles y contextos definen la habilidad de un usuario específico para hacer algo en alguna página. Los ejemplos más comunes son los roles de estudiante y maestro en el contexto de un curso, pero también existe el de administrador, que posee todos los permisos disponibles en la plataforma.

De esta forma, Moodle consigue proporcionar todas las herramientas para crear una aplicación web de educación virtual.

## **2.3. Juego de la caza del tesoro**

La caza del tesoro es un tipo de juego que lleva existiendo desde mucho antes de que la tecnología fuese capaz de ofrecerlo. Este juego ya se empleaba como técnica para formar nuevos *Scouts*, y su base está incluida en el libro “Escultismo para muchachos” de Robert Baden-Powell, el fundador del movimiento Scout [3].

El juego consiste en encontrar objetos o lugares ocultos siguiendo una serie de pistas configuradas previamente por el organizador. Por sus características, pueden ser jugados tanto en espacios interiores como en espacios exteriores. El modo de juego también puede cambiar, aceptando su realización individual, en grupos, o en modo competición. El juego permite recrear distintas rutas o caminos a seguir, compuestos de una serie de etapas. Cada etapa está formada por una localización en el mapa y una o varias pistas que ofrecen la descripción para encontrar la siguiente etapa del camino. Además, es posible configurar condiciones extras a la resolución de una etapa para poder avanzar en el juego. Al superar la etapa final, el jugador se encuentra con el “tesoro”, momento en el que se acaba el juego.

## **2.4. Aplicaciones similares existentes**

En la actualidad existen diversas aplicaciones que, sin estar todas ellas enfocadas específicamente en adaptar la caza del tesoro al mundo tecnológico y sin estar acopladas a una plataforma de gestión de aprendizaje como Moodle, aportan algunas o todas las funcionalidades necesarias que permiten construir una caza del tesoro. Varias de ellas se presentan a continuación.

### **2.4.1. Treasure-HIT**

“Treasure-HIT” es una plataforma creada por el Instituto Tecnológico de Holon (HIT), Israel, diseñada principalmente para proporcionar a los profesores un entorno de creación del juego de la caza del tesoro que apoye las actividades de aprendizaje al aire libre. La plataforma permite a los profesores crear y compartir actividades a través de su web<sup>1</sup>. Cada actividad está formada por una serie de estaciones que están compuestas por una localización, definida utilizando las funciones de Google Maps y Google Street, y una o más pistas en distintos formatos (texto, imágenes y video) que indican a los jugadores como llegar a cada estación. Los profesores también pueden adjuntar tareas a cada estación con el objetivo de ser resueltas por los jugadores como requisito para continuar avanzando en el juego.

---

<sup>1</sup> [www.treasure-hit.net](http://www.treasure-hit.net)

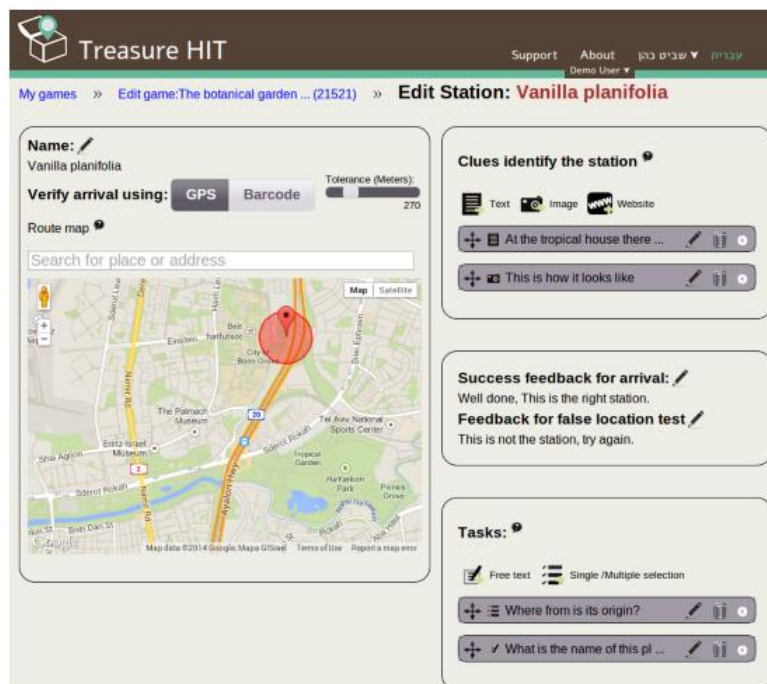


Figura 2.1 Página de configuración de una estación en Treasure-HIT

Por su parte, los alumnos deben descargarse la aplicación móvil en sus *smartphones* y acceder a la actividad concreta mediante un identificador proporcionado previamente por el profesor. Una vez en la actividad, se presentan las instrucciones iniciales sobre el juego y se indica el punto inicial, dónde se proporcionan las pistas para avanzar. Una vez que el jugador asume que se ha llegado a la posición deseada, puede verificarlo mediante un botón de “Comprobar ubicación”, que empleando la tecnología GPS disponible en el terminal móvil envía las coordenadas del usuario al servidor dónde comprueba si el jugador está dentro del rango de la estación a descubrir. Una vez superada una estación se le muestran las pistas para llegar a la siguiente hasta completar la actividad.



Figura 2.2 Interfaz de usuario de Treasure-HIT

La plataforma dispone de varios modos de juego: Como competición individual o por equipos, o como actividad a realizar individualmente por los estudiantes en su tiempo

libre [4]. Su principal inconveniente es que el código solo es accesible por sus desarrolladores, y el acceso a la plataforma debe ser validado por ellos o mediante pago.

## 2.4.2. GeoCaching

“Geocaching” es una aventura al aire libre donde los jugadores usan la aplicación móvil o un dispositivo GPS para encontrar contenedores (o también llamados *geocaches*) ocultos hábilmente en todo el mundo. Los usuarios que esconden un *geocache* apuntan las coordenadas de latitud y longitud, y luego las publican en el sitio web<sup>2</sup> para ser descubiertas por otros jugadores. Un *geocache* típico consiste en una pequeña caja física a prueba de agua como la mostrada en la Figura 2.3 que puede estar oculta en cualquier parte del mundo. Suele contener un “libro de registro” que debe ser firmado por los jugadores que se encuentran el *geocache* y, potencialmente, un “tesoro”, dependiendo del tamaño de la caja. El “tesoro” es por lo general un objeto de bajo valor, tales como pequeños juguetes y monedas. Generalmente, cuando los participantes encuentran la caja deben intercambiar tesoros - si cogen algo del contenedor deben dejar algo de similar valor. Una variante del *geocache* típico es el *multigeocache*, en el que los participantes tienen que encontrar varios *geocaches* anidados hasta encontrar el último que contiene el libro de registro. Un segundo tipo de *geocache* es el "rompecabezas". En este *geocache* los jugadores deben resolver un rompecabezas con el fin de descubrir las coordenadas del contenedor.



Figura 2.3 Contenedor de un *geocache* típico

El sitio web contiene la lista de *geocaches* disponibles en todo el mundo. Además de las coordenadas, la página web contiene información contextual acerca del contenedor, como los mapas de puntos de partida, pistas y acertijos para detectar las coordenadas, y consejos útiles o cosas a tener en cuenta. Los jugadores que decidan participar en “Geocaching” pueden ir a la página web, encontrar los cachés más cercanos en su entorno y empezar a jugar; no existe competencia, sólo la satisfacción personal [5]. Su principal inconveniente reside en que hay que ir físicamente al punto en cuestión y su preparación es muy costosa, no diferenciándose mucho del formato no tecnológico.

---

<sup>2</sup> [www.geocaching.com](http://www.geocaching.com)

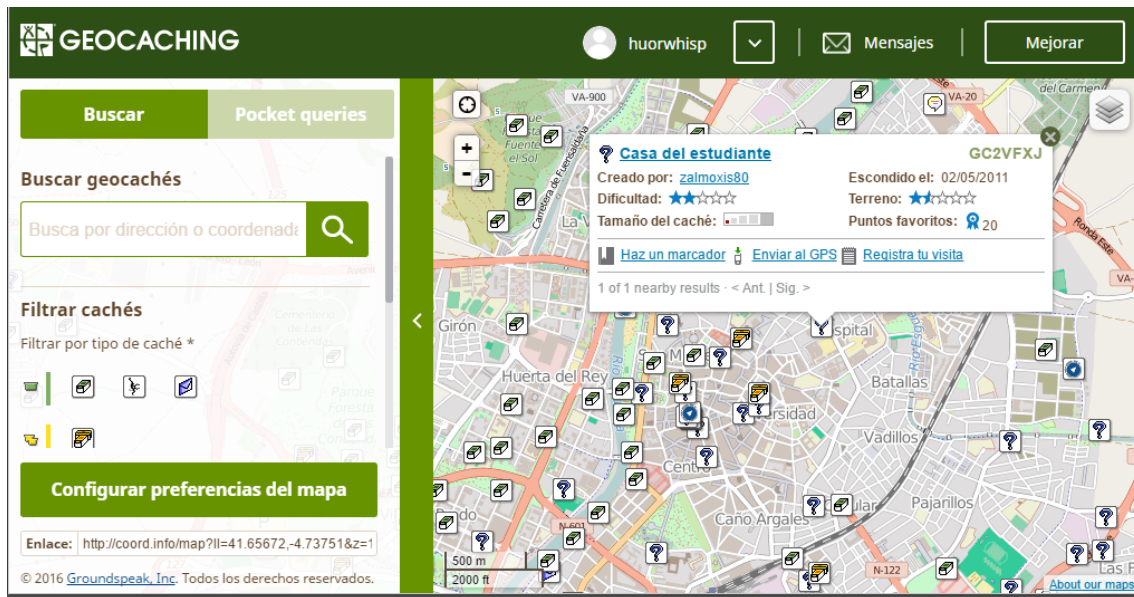


Figura 2.4 Interfaz de usuario de Geocaching

### 2.4.3. 7Scenes

“7scenes” es una plataforma comunitaria para múltiples usuarios diseñada para la creación de rutas y juegos basados en la localización. Es accesible desde su web o desde su app móvil. Proporciona un creador de escenas que permite subir contenido como fotos, videos, notas, diapositivas, retos, etc. sobre puntos concretos del mapa. También soporta muchas opciones de configuración, como la utilización de roles, niveles de juego, tiempos límite para completar la escena, navegación en el interior mediante el empleo de códigos QR, uso de mapas personalizados y formatos de juego en equipo, individual, mediante roles, etc. Esto permite a los artistas, escuelas, museos y otros desarrollar recorridos móviles interactivos y juegos sin necesidad de instalar o programar nada.

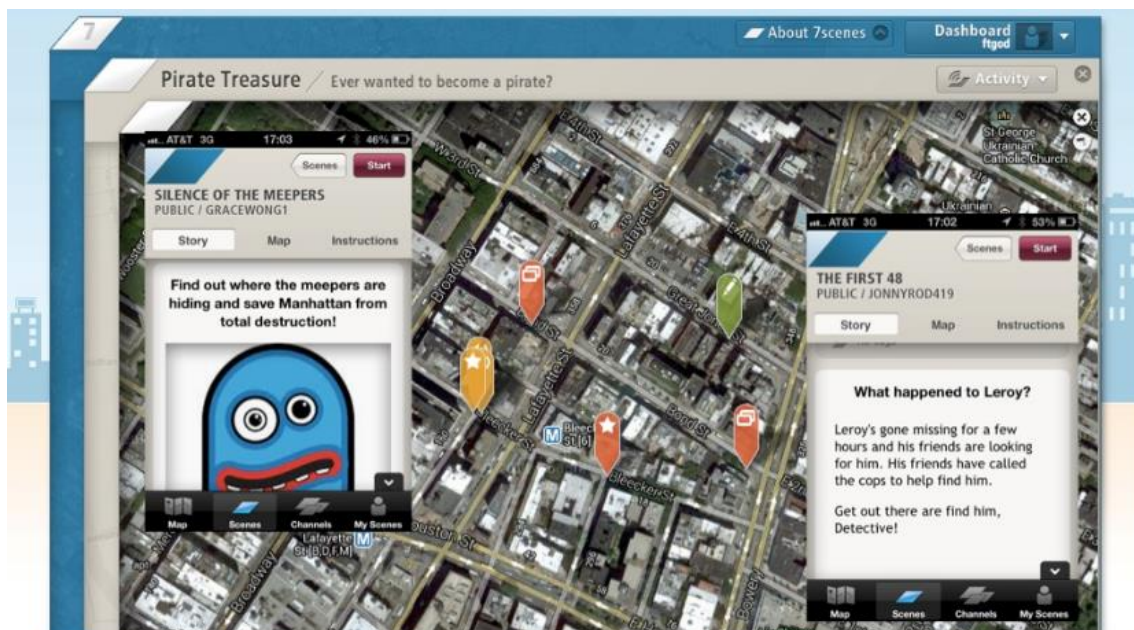


Figura 2.5 Interfaz de usuario de 7Scenes

Esta plataforma proporciona muchos tipos de juegos para los ajustes pedagógicos; y "La búsqueda del tesoro" es uno de ellos. Permite a los administradores de la escena combinar rutas a través de su entorno físico y, sobre la base de diferentes plantillas de juego con reglas específicas, los jugadores pueden añadir fotos, vídeos, audio, notas y tareas a los lugares con los que pueden ir desarrollando la historia. Los estudiantes pueden crear posteriormente una competición mediante la creación de equipos, navegar por la ciudad, descubrir la historia, colaborar con otros jugadores, realizar tareas y ganar puntos. Todo el juego es transmitido en vivo por la web y archivado para que otros jugadores (y profesores) puedan reproducir todas las interacciones y reflexionar sobre ellas con otros jugadores [6]. Su principal inconveniente reside en que no es un entorno dedicado a la realización de cazas del tesoro, por lo que al crear la ruta todas las etapas se muestran en el mapa, sin tener un control sobre cual debe mostrarse a continuación.

#### 2.4.4. Skattjat

Skattjat (Búsqueda del Tesoro en sueco) es un juego destinado a desarrollar habilidades informales de lectura de mapas y a enriquecer el aprendizaje con la historia local. En Skattjat los participantes son divididos en hasta 6 equipos equipados con un teléfono móvil. Su tarea es encontrar los puntos de referencia y recibir pistas en formato de texto o audio a través del teléfono móvil navegando mediante el uso de un mapa diseñado específicamente para el juego. Skattjat ha sido concebido y desarrollado para fomentar la actividad física en los jóvenes mediante la resolución de un misterio que rodea un castillo ubicado en el campus de la Universidad Linnaeus en Suecia. El juego está inspirado en las ideas detrás de las actividades de la búsqueda del tesoro y el deporte de orientación – un deporte de correr tradicional Escandinavo - que involucra la navegación con un mapa y una brújula [7]. Su principal inconveniente es que es un juego dedicado únicamente a una actividad, que no proporciona un entorno de creación de nuevas actividades.



Figura 2.6 Interfaz de usuario de Skattjat

#### 2.4.5. fAR-Play

“fAR-Play”, nombrado así como acrónimo de *Augmented Reality Play* (Realidad de Juego Aumentada), es un *framework* creado por la Universidad de Alberta destinado a

desarrollar juegos de realidad aumentada. Permite utilizar la plataforma para colocar puntos de interés virtuales (animales, estatuas, postes de señales, etc.) en lugares del mundo real. Estos puntos se muestran a los usuarios en una imagen superpuesta sobre los objetos físicos cerca de ellos. A continuación, un Layar (capa de realidad aumentada creada sobre una localización a través de la app del mismo nombre) es creado para permitir a los jugadores localizar los puntos de interés al aire libre, utilizando un teléfono inteligente (los puntos de interés interiores son escaneados utilizando etiquetas QR). “fAR-Play” da la capacidad de crear dos tipos de juego: Por un lado, una búsqueda del tesoro virtual, dónde los puntos de interés son colocados libres en un área dada y los jugadores los capturan y coleccionan en cualquier orden. Por otro, una cacería del tesoro virtual, que permite una carrera basada en objetivos. Los puntos de interés no son de libre detección, sino que deben ser descubiertos en orden. Cada punto de interés, al ser escaneado, planteará una pregunta de selección múltiple. Al responder correctamente se le dará al jugador (s) una pista sobre el siguiente punto de interés. El primer equipo que escanea el punto de interés final, gana el juego [8]. Su principal inconveniente reside en que no proporciona un entorno de creación de actividades a los nuevos usuarios de la plataforma.



*Figura 2.7 Ejemplo de imagen mostrada con realidad aumentada en fAR-Play*

## 3. Análisis del problema

### 3.1. Introducción

Vistas las desventajas que ofrecen las principales aplicaciones del tipo “La caza del tesoro”, si un profesor se propone la realización de una actividad de este tipo tendría que escribir a ordenador (o en su defecto usar papel y boli) cada pista de cada etapa del camino, imprimirlas e ir físicamente a cada localización para colocarlas. Como es comprensible, esto supone una serie de desventajas significativas:

- ❖ El tiempo empleado por el profesor tanto en crear la caza como en desplazarse a cada lugar es muy alto. El coste puede provocar el desánimo y el abandono de crear este tipo de actividades.
- ❖ Las condiciones meteorológicas o las propias personas pueden destruir o alterar las pistas colocadas, haciendo imposible la realización de la caza a partir de ese punto y sometiendo a la actividad a un mantenimiento y control por parte del profesor nada desdeñable.
- ❖ Existe un gasto en papel que podría ser evitado.
- ❖ Debe proporcionarles en mano a todos los alumnos la primera pista.
- ❖ Cuando comienza la caza, el profesor no tiene un seguimiento en tiempo real del progreso de los usuarios.
- ❖ Empleando este sistema tradicional es muy trabajoso poner requisitos previos que deben cumplirse para desvelar una pista. Para ello, tendría que especificar que deben realizar una actividad completa y el propio profesor deberá supervisar su realización para luego entregarles en mano la pista bloqueada.
- ❖ No existe un registro de los errores que han cometido los jugadores en las deducciones de las pistas y que les han llevado a una mala ubicación.
- ❖ Desde el punto de vista de los jugadores, no existe un sistema de revisión que permita ver el recorrido y los intentos realizados.
- ❖ El único método de calificación posible es por posición al completar la caza.

Este trabajo busca desarrollar una adaptación al plano tecnológico que solucione todas las desventajas expuestas, y además añade funcionalidades que solo pueden conseguirse utilizando una plataforma educativa como Moodle.

### 3.2. Requisitos del trabajo

Para adaptar la caza del tesoro al plano tecnológico, se imponen una serie de requisitos que deben garantizarse para lograr una funcionalidad plena. Se va a distinguir entre los requisitos globales que deben garantizarse en el módulo, así como los requisitos específicos de configuración del juego manejados por el profesor.



### 3.2.1. Requisitos globales del módulo

- Deberá estar disponible en su versión inicial en los idiomas de español e inglés.
- Deberá cumplir las exigencias de compatibilidad con todas las bases de datos que soporta Moodle.
- Deberá incorporar una interfaz intuitiva y fácil de usar de creación y edición de las geometrías asociadas a cada etapa.
- Deberá incorporar una interfaz de juego a pantalla completa, apta, intuitiva y atractiva para dispositivos móviles.
- Deberá registrar todos los intentos que realiza un usuario/grupo.
- Cada usuario o grupo deberá poder ver todos sus intentos realizados.
- Deberá permitir ver al profesor los intentos realizados de todos los usuarios/grupos.
- Deberá implementar un cuadro de información con el progreso de todos los alumnos/grupos accesible tanto por los profesores como por los alumnos para fomentar la competitividad.
- Deberá existir al menos un camino por cada instancia de la actividad.
- Cada camino deberá incorporar al menos dos etapas y que todas las etapas tengan una geometría asignada para que el camino sea válido.
- Cada etapa deberá permitir asignar una actividad a completar y/o una pregunta simple a resolver antes de que se le muestre al usuario la pista asociada a la etapa.
- Deberá informar de los caminos no validados, así como de los alumnos que no tienen ningún camino asignado o que tienen asignado más de uno.
- Deberá informar de todos los errores que se produzcan durante el desarrollo del juego o de la edición.
- Deberá permitir la creación de varias localizaciones para cada etapa. Esta opción se hace necesaria en aquellas etapas que pueden ser validadas desde distintos lugares.
- No deberá permitir eliminar ni cambiar el orden de las etapas asignadas a un camino una vez que un primer usuario haya realizado un intento sobre ese camino.

### 3.2.2. Requisitos de configuración del juego

- El profesor deberá poder elegir entre diferentes métodos de calificación, con una nota máxima asociada.
- El profesor deberá poder asignar un porcentaje de penalización por fallo en la localización o respuesta a una pregunta enviadas por un alumno.

- El profesor deberá poder escoger si la actividad se realiza en modo grupo o individual.
- El profesor deberá poder seleccionar si la actividad se realiza acudiendo presencialmente a las geometrías de cada etapa o bien mediante una marca sobre el mapa que indica la localización que el usuario desea enviar.
- El profesor deberá poder escoger la fecha de inicio de la actividad y fecha de cierre, a partir de la cual no se pueden enviar más intentos.

## 4. Desarrollo del proyecto

La presente sección pretende explicar las decisiones que se han tomado para la creación y la puesta en marcha del módulo de actividad georreferenciado para Moodle. Con el fin de esclarecer los matices de la solución, la explicación se ha dividido en dos aspectos fundamentales: los técnicos, que incluyen el desarrollo de todos los parámetros necesarios para poder empezar a escribir el código del módulo, y los funcionales, que incluyen todas las soluciones que se han aportado para crear cada funcionalidad del módulo.

### 4.1. Aspectos técnicos

#### 4.1.1. Configuración del entorno de trabajo

Antes de comenzar con el desarrollo del código, es importante definir el entorno de trabajo necesario para crear y depurar de la forma más eficiente el nuevo módulo para Moodle. Para ello se han seguido una serie de pautas:

1. Se ha descargado la más reciente versión de Moodle disponible (la 2.9.2 en la fecha de comienzo) desde la página oficial [9]. De esta forma se puede desarrollar el módulo en la plataforma más actual con todas las nuevas opciones de desarrollo disponibles.
2. Se han investigado los requisitos mínimos para poner en funcionamiento la versión de Moodle descargada [10]. Entre ellos, la plataforma exige una versión mínima del servidor PHP de 5.4.4 y utilizar una de las siguientes bases de datos soportadas: PostgreSQL (versión mínima 9.1), MySQL (versión mínima 5.5.31), MariaDB (versión mínima 5.5.31), Microsoft SQL Server (versión mínima 2008) y Oracle Database (versión mínima 10.2).
3. Se han investigado las principales plataformas que proporcionan la estructura para crear el servidor web. Dado que el sistema operativo del ordenador dónde se va a desarrollar el módulo es Windows 10, se ha decidido descargar e instalar el paquete XAMPP [11]. XAMPP es una distribución de Apache disponible para las plataformas Windows, Linux y OS X que para la versión descargada contiene MariaDB (v10.1.8) y PHP (v5.6.14) entre otros, características que sobrepasan los requisitos mínimos para funcionar de la versión de Moodle elegida.
4. Se ha creado la base de datos y el usuario que se utilizará para los accesos a la misma en Moodle. También el directorio *moodldata*, dónde se almacenarán los ficheros subidos a la plataforma. Luego, a partir del fichero *'config-dist.php'* en el directorio principal de Moodle se ha creado el fichero de configuración con los parámetros básicos, que incluyen los atributos para acceder a la base de datos, la dirección web principal del sitio y la localización elegida del directorio *moodldata*.
5. Se ha buscado un entorno de desarrollo que soporte la creación de código PHP, el lenguaje usado para el desarrollo de la plataforma Moodle, pero también maneje JavaScript, necesario para otorgar fluidez en ciertas operaciones en el

cliente. Finalmente se ha optado por el entorno de desarrollo integrado Netbeans [12], que permite crear código en los lenguajes de programación mencionados de forma simple, además de aportar otras funcionalidades.

6. Se ha configurado Xdebug [13], una extensión php que permite realizar depuración del código ejecutado desde el editor, tanto en el servidor web como en Netbeans. También se ha descargado el *plugin* para Chrome (el navegador principal que se usará para depurar) Xdebug Helper que permite habilitar o deshabilitar la sesión de depuración desde el navegador.
7. Se ha creado una cuenta en la plataforma GitHub [14] y un repositorio dónde se almacenará el código desarrollado del módulo. A su vez, se ha descargado la plataforma de gestión de versiones Git que junto con la interfaz gráfica que proporciona TortoiseGit permitirá realizar un manejo del control de versiones del código de forma sencilla.

#### 4.1.2. Directrices del código

Moodle es un proyecto colaborativo que involucra a centenares de desarrolladores en todo el mundo. Por ello la plataforma establece ciertas recomendaciones a la hora de añadir o modificar código para que la consistencia y estabilidad del proyecto sean fuertes [15]. Las siguientes secciones definen las pautas que se han seguido con la mayor exactitud en la creación del módulo de la caza del tesoro.

##### 4.1.2.1. Reglas Generales

Las reglas generales para cualquier ampliación o modificación de la plataforma Moodle son las siguientes:

1. Todos los archivos de texto deberían utilizar el formato de texto Unix (la mayoría de los editores de texto tienen esto como una opción).
2. Todas las etiquetas php deben ser 'completas' como `<?php ?>` ... no 'reducidas' como `<? ?>`.
3. Todos los avisos de copyright deben ser mantenidos.
4. Cualquier include/require debería utilizar una ruta absoluta que comience por `$CFG->dirroot` o `$CFG->libdir`, nunca relativos, ya que estos en algunas ocasiones funcionan de forma extraña en PHP.
5. Cada archivo debería comprobar que el usuario está autenticado correctamente, utilizando las funciones `require_login()` y `isadmin()`, `isteacher()`, `iscreator()` o `isstudent()`.
6. 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. Si se quiere escribir código SQL entonces se deberá comprobar que: funciona en cualquier plataforma; restringido a funciones

específicas del código (normalmente un archivo lib.php); y claramente comentado.

7. No crear o utilizar variables globales distintas de las estándar \$CFG, \$SESSION, \$THEME, \$SITE, \$COURSE y \$USER.
8. Todas las variables deberían ser inicializadas o, al menos, comprobada su existencia utilizando `isset()` o `empty()` antes de ser utilizadas.
9. Todas las cadenas deberían ser traducibles.
10. Todos los errores deberían ser visualizados utilizando la función `print_error()` para maximizar la traducción y ayudar a los usuarios.
11. Todos los ficheros de ayuda deben ser traducibles.
12. La información que llega desde el navegador (enviada con los métodos GET o POST) automáticamente tiene las "magic\_quotes" aplicadas (sin importar la configuración de PHP) por lo que puede ser insertada con total seguridad en la base de datos. El resto de la información (obtenida desde los archivos, o desde la base de datos) debe ser escapada con la función `addslashes()` antes de insertarla en la base de datos.
13. 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 grabadas utilizando los eventos de la actividad.
15. Los enlaces HTML deben ser generados siempre relativos a la raíz del sitio Moodle.

#### 4.1.2.2. Estilo del código

El estilo de código en un proyecto colaborativo en el que están involucrados muchos desarrolladores debe seguir un estándar que permita entender y modificar el código a cualquier programador que quiera realizar una nueva versión o arreglo, por ello la plataforma Moodle define las pautas más importantes:

1. El sangrado del texto debe ser siempre de 4 espacios. No se deben utilizar los tabuladores nunca.
2. Los nombres de las variables tienen que ser siempre fáciles de leer, procurando que sean palabras en minúsculas con significado en inglés. Si realmente se necesita más de una palabra, se deben poner juntas, pero procurando que sean tan breves como sea posible. Se deben utilizar nombres en plural para las matrices de objetos.

3. Las constantes tienen que definirse siempre en mayúsculas, y empezar siempre por el nombre del módulo al que pertenecen. Deberían tener las palabras separadas por guiones bajos.
4. Los nombres de las funciones tienen que ser palabras sencillas en minúsculas y en inglés, y empezar con el nombre del módulo al que pertenecen para evitar conflictos entre módulos. Las palabras deberían separarse por guiones bajos. Los parámetros, si es posible, tendrán valores por defecto. Se debe comprobar que no haya espacio entre el nombre de la función y lo siguiente (paréntesis).
5. Los bloques de código siempre deben estar encerrados por llaves (incluso si solo constan de una línea).
6. Las cadenas tienen que ser definidas utilizando comillas simples siempre que sea posible, para obtener un mejor rendimiento.
7. 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 en línea deberían utilizar los caracteres //, alineados con cuidado por encima de las líneas de código que comenta.
8. El espacio en blanco se puede utilizar con bastante libertad para ganar claridad. Generalmente, debería haber un espacio entre llaves y líneas normales y ninguno entre llaves y variables o funciones.
9. Cuando se esté realizando una copia de un objeto, se debe utilizar siempre la función `clone()` originalmente sólo disponible en php5 (en caso contrario simplemente tendrá una referencia al primer objeto). Moodle garantiza que este método funcionará también bajo php4. Si lo que se quiere copiar no es un objeto, pero puede contener objetos (p.ej. un array de objetos) utilice la función `fullclone()` en su lugar.

#### 4.1.2.3. Estructura de la base de datos

Al igual que con el estilo de código es necesario tener en cuenta una serie de pautas a la hora de desarrollar la estructura de la base de datos para Moodle:

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 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 un módulo que contiene información sobre 'cosas', deberían ser llamadas `módulo_cosas`.

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, módulo) debería ser llamado `móduloid`.
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. Es conveniente definir siempre un valor por defecto para cada campo.
10. Cada tabla debe comenzar con el prefijo de la base de datos (`$CFG->prefix`).
11. Reglas 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 entrecomillar el valor.

#### 4.1.2.4. Normas de seguridad

Esta sección especifica una serie de pautas que se deben seguir antes de escribir nuevo código para Moodle con el fin de evitar posibles vulnerabilidades de seguridad:

1. No basarse en `register_globals`. Cada variable debe ser correctamente inicializada.
2. Inicializar todos los arrays y objetos, aunque estén vacíos.
3. No utilizar la función `optional_variable()`. En su lugar, utilizar la función `optional_param()`.
4. No utilizar la función `require_variable()`. En su lugar, utilizar la función `required_param()`.
5. Utilizar `data_submitted()` con cuidado. La información debe ser limpiada antes de utilizarla.
6. No utilizar `$_GET`, `$_POST` o `$_REQUEST`. En su lugar, utilizar las funciones `required_param()` u `optional_param()`.

7. No comprobar las acciones con código como: `if (isset($_GET['algo']))`.
8. Cuando sea posible agrupar todas las llamadas a `required_param()`, `optional_param()` y el resto de inicialización de variables en el principio de cada fichero.
9. Utilizar el mecanismo `sesskey` para proteger el envío de formularios de ataques.
10. Todos los nombres de ficheros deben ser 'limpiados' utilizando la función `clean_filename()`.
11. Cualquier información leída desde la base de datos debe tener la función `addslashes()` aplicada antes de volver a enviar la información a la base de datos.
12. La información que se almacenará en la base de datos debe venir de peticiones POST.
13. No utilizar información obtenida de `$_SERVER`.
14. La información enviada a la base de datos debe ser filtrada mediante la función `clean_param()`.
15. Asegurarse de que el código SQL es correcto.
16. Comprobar toda la información (especialmente la que es enviada a la base de datos) en cada archivo que es utilizada.
17. Los bloques de código que se incluyan deben presentar una estructura PHP correcta.
18. Para utilizar funciones que invoquen un Shell hay que asegurarse de que se han limpiado los parámetros anteriormente con `escapeshellcmd()` o `escapeshellarg()`.

### 4.1.3. Estructura de ficheros del módulo

Todos los módulos de actividad de Moodle están divididos en directorios del mismo nombre que el módulo en cuestión y residen dentro del subdirectorio `/mod` de la carpeta principal de Moodle. El contenido de estos directorios lo forman una serie de ficheros, algunos de ellos obligatorios para cualquier módulo de actividad y otros puestos específicamente por el desarrollador para otorgar características únicas a la aplicación docente.

Para facilitar la creación de un nuevo módulo al desarrollador, Moodle proporciona una plantilla con los directorios y ficheros básicos accesible desde [16] y que está adaptada para las versiones 2.x de la plataforma.

A continuación, se detalla la estructura de ficheros del módulo la caza del tesoro y se explica qué función tiene cada uno de ellos.



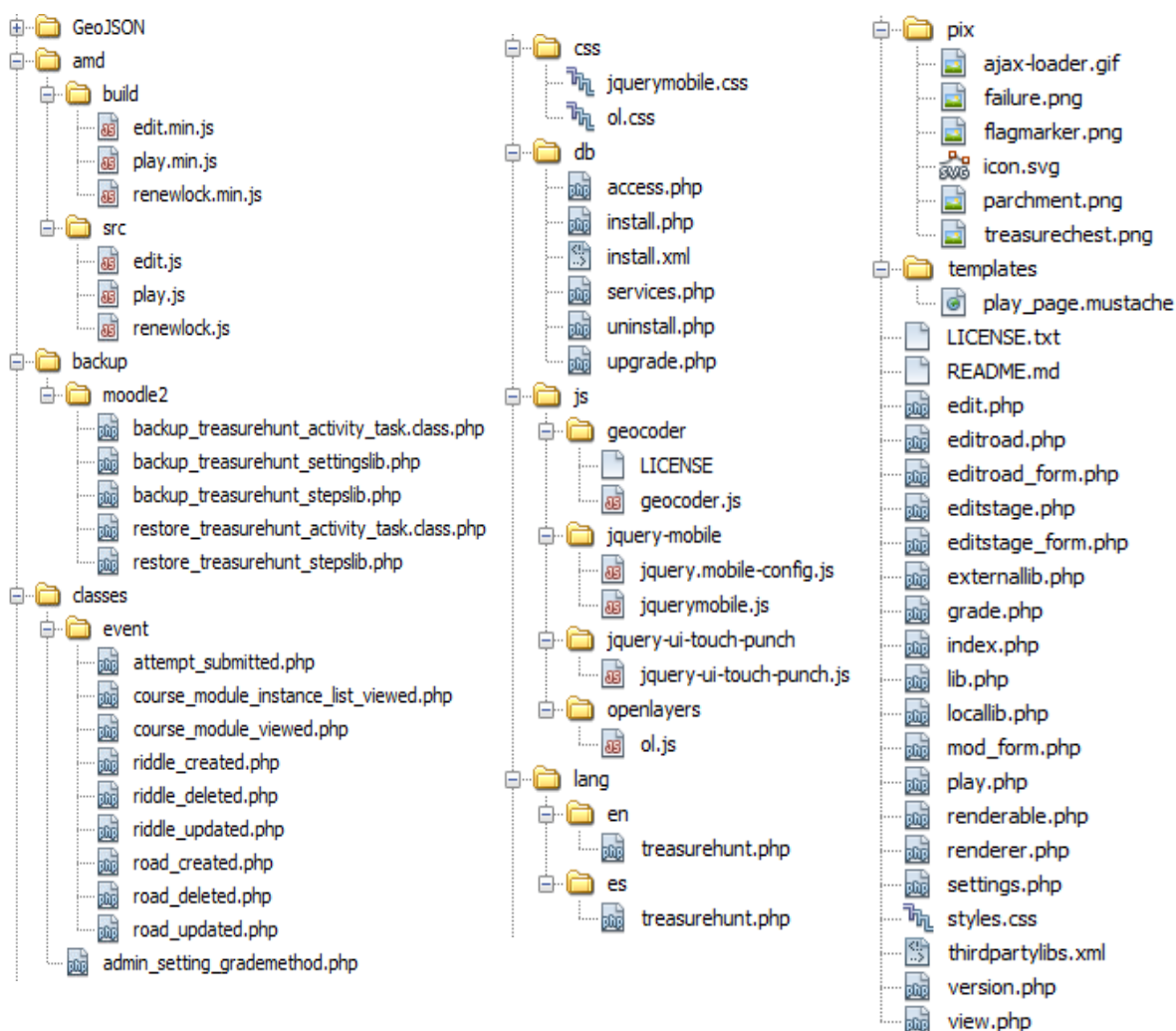


Figura 4.1 Estructura de ficheros del módulo la caza del tesoro.

Ficheros que son imprescindibles para la integración completa con Moodle:

- ❖ **Directorio backup:** Este directorio agrupa los ficheros encargados de definir cómo hacer un respaldo o recuperación de la actividad. Existen múltiples formatos para realizar un *backup* en Moodle, pero el más actual y recomendado es ‘*moodle2*’, que es el requerido para mantener cualquier módulo o bloque transportable entre instancias. En un directorio del mismo nombre que el formato mencionado se encuentran los siguientes ficheros:
  - **Fichero *backup\_treasurehunt\_activity\_task.class.php*:** Se encarga de definir la tarea de respaldo mediante la llamada a los pasos a ejecutar. También codifica los URL para poder restaurarlos de manera adecuada.
  - **Fichero *backup\_treasurehunt\_settingslib.php*:** Se encarga de almacenar todas las opciones personalizadas para realizar la copia de seguridad. Para el módulo de la caza del tesoro no es necesario crear ninguna opción adicional a las de por defecto, pero el fichero se mantiene por si hay futuros cambios.

- ***Fichero backup\_treasurehunt\_stepslib.php***: Se encarga de implementar los pasos que son llamados desde la tarea para hacer una copia de seguridad de toda la actividad desde la base de datos.
- ***Fichero restore\_treasurehunt\_activity\_task.class.php***: Se encarga de definir la tarea de restauración mediante la llamada a los pasos a ejecutar. También decodifica a su vez los URL codificados para que funcionen correctamente.
- ***Fichero restore\_treasurehunt\_stepslib.php***: Se encarga de implementar los pasos contrarios a los que realiza la copia de seguridad, guardando los nuevos datos de la actividad en la base de datos.

❖ **Directorio classes:**

- ***Directorio events***: Este directorio agrupa todos los ficheros que se encargan de definir las clases con los eventos que pueden producirse durante la ejecución de una actividad. Un evento en Moodle se define como una pieza atómica de información que describe algo que sucedió y que consta de un nombre, una descripción y opcionalmente una URL que dirige al punto dónde se puede observar el resultado del evento. Para lanzar un evento solo es necesario crear una instancia de la clase y aplicar el método `trigger()`. Para el módulo de la caza del tesoro se crean los siguientes ficheros:
  - Fichero *'attempt\_submitted.php'*: Define el evento que comunica el envío de un nuevo intento por parte de un usuario.
  - Fichero *'course\_module\_instance\_list\_viewed.php'*: Define el evento a nivel del curso que comunica que un usuario ha visto la página que muestra todas las cazas del tesoro del curso.
  - Fichero *'course\_module\_viewed.php'*: Define el evento a nivel del curso que comunica que un usuario ha visto la pantalla principal del módulo.
  - Fichero *'stage\_created.php'*: Define el evento que comunica que un usuario ha creado una nueva etapa.
  - Fichero *'stage\_updated.php'*: Define el evento que comunica que un usuario ha actualizado una etapa.
  - Fichero *'stage\_deleted.php'*: Define el evento que comunica que un usuario ha eliminado una etapa.
  - Fichero *'road\_created.php'*: Define el evento que comunica que un usuario ha creado un nuevo camino.
  - Fichero *'road\_updated.php'*: Define el evento que comunica que un usuario ha actualizado un camino.

- Fichero *'road\_deleted.php'*: Define el evento que comunica que un usuario ha eliminado un camino.
- **Fichero *admin\_setting\_grademethod.php***: Define la clase que permite seleccionar el método de calificación en las opciones de administración del módulo de la caza del tesoro (fichero *'settings.php'* que se explica más adelante).

❖ **Directorio db:**

- **Fichero *access.php***: Se encarga de definir los permisos (capacidades) disponibles en el módulo. A cada permiso se le asignan los roles básicos de Moodle (profesor, estudiante, gestor...) que lo soportan, es decir, que pueden emplear esa capacidad dentro del módulo, como puede ser crear una nueva etapa, jugar la actividad, etc. Aunque aquí se establezcan las capacidades que soporta cada rol, el administrador de Moodle puede modificar estas asignaciones a su gusto.

Para el módulo de la caza del tesoro se han implementado los siguientes permisos:

- *'addinstance'*: Especifica qué roles de Moodle pueden crear una instancia del módulo. Se otorga este permiso a los roles de *'profesor con derecho a edición'* y *'administrador'*.
- *'view'*: Especifica qué roles pueden ver la pantalla principal de la actividad. Se otorga este permiso a los roles de *'invitado'*, *'estudiante'*, *'profesor'*, *'profesor con derecho a edición'* y *'administrador'*.
- *'play'*: Especifica qué roles pueden jugar la actividad. Se otorga este permiso al rol de *'estudiante'*.
- *'managetreasurehunt'*: Especifica qué roles pueden entrar en la pantalla de edición de la actividad. Se otorga este permiso a los roles de *'profesor con derecho a edición'* y *'administrador'*.
- *'editroad'*: Especifica qué roles pueden editar los atributos de un camino o eliminarlo. Se otorga este permiso a los roles de *'profesor con derecho a edición'* y *'administrador'*.
- *'editstage'*: Especifica qué roles pueden editar los atributos de una etapa o eliminarla. Se otorga este permiso a los roles de *'profesor con derecho a edición'* y *'administrador'*.
- *'addroad'*: Especifica qué roles pueden añadir un nuevo camino. Se otorga este permiso a los roles de *'profesor con derecho a edición'* y *'administrador'*.
- *'addstage'*: Especifica qué roles pueden añadir una nueva etapa. Se otorga este permiso a los roles de *'profesor con derecho a edición'* y *'administrador'*.

- ‘viewusershistoricalattempts’: Especifica qué roles pueden ver el historial de intentos de un usuario o grupo concreto. Se otorga este permiso a los roles de ‘profesor’, ‘profesor con derecho a edición’ y ‘administrador’.

Una vez configurados los permisos de acceso, o también llamados capacidades, se puede comprobar si un usuario dispone de un permiso en concreto con la función `has_capability()`, que devuelve verdadero o falso, pudiendo definir que hacer a continuación. Si además se quiere garantizar que a una pantalla no se pueda acceder sin que el usuario actual tenga una capacidad concreta, existe la función `require_capability()`. De esta forma, la plataforma permite diferenciar lo que se le ofrece en cada pantalla al usuario dependiendo del rol que ocupa en ese contexto dado.

- **Fichero *install.php***: Se encarga de definir el código a ejecutar justo después de crear el esquema de la base de datos del módulo del fichero ‘*install.xml*’. Se usa principalmente para comprobar restricciones impuestas al módulo o instalar algún requerimiento especial. Para el módulo de la caza del tesoro no es necesario utilizarlo, pero el fichero se mantiene por si hay futuros cambios.
- **Fichero *install.xml***: Se encarga de definir la estructura de las tablas, así como las columnas específicas y propiedades que las componen que se crearán en la base de datos durante la instalación del módulo en Moodle. Para ello emplea la capa de abstracción XMLDB diseñada para garantizar la compatibilidad entre todas las bases de datos soportadas por Moodle. Una vez instalado no se recogerán los cambios producidos en este fichero y para realizar cualquier modificación en la base de datos será necesario utilizar el fichero ‘*upgrade.php*’ explicado a continuación. En el apartado 4.1.4 se explicará el estudio de las tablas y columnas elegidas.
- **Fichero *upgrade.php***: Se encarga de definir las actualizaciones que se deben llevar a cabo en la base de datos, ya sea crear, editar o eliminar una tabla o columna del esquema. Para ello, al realizarse la actualización del módulo comprueba si existen tareas de actualización entre la versión instalada y la versión a instalar. Para el módulo de la caza del tesoro no es necesario utilizarlo porque se parte de una versión 0, encontrándose todos lo relativo a la instalación de la base de datos en el fichero ‘*install.xml*’.
- **Fichero *uninstall.php***: Se encarga de definir el código a ejecutar justo después de eliminar el esquema de la base de datos del módulo del fichero ‘*install.xml*’. Se utiliza principalmente para desinstalar algún requerimiento especial. Para el módulo de la caza del tesoro no es necesario utilizarlo, pero el fichero se mantiene por si hay futuros cambios.
- **Fichero *services.php***: Se encarga de definir el nombre, la clase, la ubicación (por norma general el fichero ‘*externallib.php*’) y la descripción de las funciones del servicio web del módulo. Además, permite definir restricciones que se aplican al acceso al servicio, como la necesidad de

ciertos permisos o el bloqueo de todos los usuarios a los que no conceda autorización el administrador. En el módulo de la búsqueda del tesoro, las funciones del servicio web son implementadas para el uso de AJAX (Asynchronous JavaScript And XML) desde el cliente y así poder dotar de todas las funcionalidades de la caza del tesoro sin depender de actualizar continuamente la página, consiguiendo una interfaz más fluida.

- ❖ **Directorio lang:** Este directorio agrupa todos los ficheros dónde se guardan las variables con las traducciones a las cadenas de caracteres que se muestran en la interfaz del módulo, ya sea el nombre de un botón como un aviso de error. En los módulos, estos ficheros reciben el mismo nombre que el módulo en cuestión, y permiten distinguir cada idioma soportado mediante su introducción en subdirectorios cuyo nombre es el código del idioma asociado. Así pues, el fichero *'treasurehunt.php'* que se encuentra en el subdirectorio *'es'* contiene todas las traducciones de las cadenas en español. El formato que sigue una cadena dentro de estos directorios es el siguiente:

```
$string['example']= 'Esta es una cadena de ejemplo';
```

*Figura 4.2 Formato de una cadena de ejemplo en español.*

```
$string['example']= 'This is an example string';
```

*Figura 4.3 Formato de una cadena de ejemplo en inglés.*

Una vez definidas, se puede recuperar la cadena de texto para imprimirlo en el idioma seleccionado en la plataforma (siempre que el módulo lo soporte, si no se empleará el de por defecto, por lo general en inglés) mediante el comando:

```
get_string('example','treasurehunt');
```

Con ello se consigue asegurar que el módulo de la caza del tesoro soportará un sistema multilingüe sin que suponga un gran esfuerzo el desarrollo de un nuevo fichero de idioma, aunque por el momento solo está integrado en inglés y en español.

- ❖ **Directorio pix:** Este directorio alberga todas las imágenes necesarias de la aplicación. Entre ellas destaca el icono principal del módulo (icon.svg) que se puede apreciar en la Figura 4.4. Este está disponible en el formato recomendado por Moodle para los iconos, el SVG (Scalable Vector Graphics), que permite cambiar la escala de la imagen sin perder calidad. El icono será visible durante la creación de una instancia y en la lista de actividades de un curso, a la derecha del nombre de la actividad. Además, el resto de imágenes salvo una se han insertado en formato PNG (Portable Network Graphics) que permite aprovechar la característica de transparencia que soporta, así como su gran compresión. Estas imágenes se utilizan en la pantalla de juego para representar las marcas sobre el mapa. La imagen restante se encuentra en formato GIF (Graphics Interchange

Format) para aprovechar la característica de animación y recrear un icono que aparece en los momentos de carga de algún contenido de la pantalla.



Figura 4.4 Icono principal del módulo.

- ❖ **Fichero `index.php`:** Este fichero define la página con la tabla que enumera todas las instancias del módulo en el curso. En el módulo de la caza del tesoro la tabla consta de dos columnas, una que indica la semana del curso asignada a la actividad y otra con el nombre de cada actividad del módulo en forma de link que redirige a la instancia seleccionada.
- ❖ **Fichero `lib.php`:** Este fichero contiene el núcleo de funciones básicas que son llamadas por el sistema de Moodle para lograr la integración del módulo en la plataforma. Todas las funciones de la caza del tesoro que implementan el núcleo de Moodle comienzan por `'treasurehunt_'` y es seguido por el nombre de la acción en cuestión. Algunas de las principales que almacena este fichero son:
  - `treasurehunt_add_instance($treasurehunt)`: Añade a la base de datos una nueva instancia de la caza del tesoro. Es llamada una vez se crea una nueva actividad del módulo en un curso y se el formulario de configuración.
  - `treasurehunt_update_instance($treasurehunt)`: Actualiza el registro de la base de datos de una instancia existente de la caza del tesoro. Es llamada cuando se modifica el formulario de configuración de la actividad de la caza del tesoro asociada.
  - `treasurehunt_delete_instance($id)`: Elimina de la base de datos una instancia de la caza del tesoro. Es llamada cuando se elimina la actividad de la caza del tesoro asociada.
  - `treasurehunt_supports($feature)`: Define las características comunes soportadas por el módulo, como la capacidad de albergar calificaciones, mostrar descripciones, realizar copias de seguridad, etc.

Además, puede contener otras funciones opcionales que implementen la lógica de negocio del módulo, aunque en la caza del tesoro se ha decidido crearlas en un fichero distinto (`locallib.php`) para diferenciarlas de una manera más clara y ahorrar memoria cuando Moodle está realizando acciones en todos los módulos cargando sus ficheros `'lib.php'`.

- ❖ **Fichero `locallib.php`:** Como se ha comentado, este fichero define todas las funciones que implementan la lógica de negocio del módulo, tales como crear, actualizar o eliminar etapas y caminos, así como las que se encargan de enviar el renderizado de cada sección de una página. Desde este fichero se puede acceder a

su vez a las funciones del núcleo y a las clases que se pasarán al renderizado gracias a la conexión existente con el fichero *'lib.php'* y el fichero *'renderable.php'* que se detalla más adelante en los ficheros específicos del módulo. La nomenclatura de las funciones sigue la misma línea que la presente en el fichero *'lib.php'*.

❖ **Fichero *mod\_form.php***: Define el formulario de configuración de los parámetros básicos de la actividad. Este se muestra a los usuarios cuando se crea o se edita una instancia de la actividad. Para el caso particular del módulo de la caza del tesoro este formulario está compuesto por los siguientes campos:

- Nombre de la instancia
- Descripción de la instancia.
- Casilla para mostrar la descripción en la página del curso.
- Selector de la fecha de inicio del juego.
- Selector de la fecha de finalización del juego.
- Casilla para mostrar siempre la descripción de la instancia.
- Casilla para realizar la actividad en grupos.
- Selector del tipo de calificación.
- Selector de la categoría de la calificación
- Calificación necesaria para aprobar.
- Selector del método de calificación.
- Penalización por fallo en localización.
- Penalización por fallo en respuesta.
- Otros campos comunes a otros módulos, como visibilidad, restricciones de acceso, etc.

❖ **Fichero *style.css***: Fichero que contiene todas las opciones de estilo propias de la actividad.

❖ **Fichero *version.php***: Fichero que define la versión actual del módulo, además de otros atributos como la versión mínima requerida de Moodle o las dependencias que existen con otros módulos para poder instalarse o actualizarse. También permite definir el nivel de madurez del módulo, desde el estado *alpha* hasta el estable. En el caso concreto del módulo de la caza del tesoro se especifica que requiere la versión mínima de Moodle 2.9.0, que la última versión del módulo es la 1.0 que data de la fecha 29 de Junio de 2016, que el nivel de madurez está testeado y pasará a estable si no ocurre un error fatal y que no existen dependencias con otros módulos.

```
$plugin->component = 'mod_treasurehunt';  
$plugin->version = 2016062900; // 29/6/2016  
$plugin->release = 'v1.0';  
$plugin->requires = 2015051100; // Moodle 2.9  
$plugin->maturity = MATURITY_RC;  
$plugin->dependencies = array();
```

Figura 4.5 Contenido del fichero *versión.php*

- ❖ **Fichero *view.php***: Fichero que define la pantalla principal de un módulo. Para la caza del tesoro, existe la sección de información general, la de historial de intentos y la de vista del progreso de todos los usuarios. En función de los permisos que el usuario posea al entrar en esta pantalla se le mostrarán unas secciones u otras.

Una vez explicados los directorios y ficheros comunes para el funcionamiento de un módulo en Moodle, se explica a continuación los utilizados particularmente por la caza del tesoro para otorgarle su funcionalidad única:

- ❖ **Directorio *GeoJSON***: Dentro de este directorio se encuentra la librería *GeoJSON* que forma parte del framework *MapFish* [17]. Esta librería está implementada en PHP y compuesta de todas las clases que permiten crear y modificar *features* además de colecciones de estas. Una *feature* en el argot de los sistemas de información geográfica se define como un objeto que tiene atributos espaciales (la geometría que utiliza, ya sean puntos, líneas, polígonos o multipolígonos con su ubicación en el plano) y atributos no espaciales que dicen algo sobre esa geometría, como su nombre, descripción, etc. También permite transformar indistintamente las *features* a formato *GeoJSON* [18], y adaptar las geometrías que las componen a formato *WKT* (Well Known Text) [19]. En el módulo de la caza del tesoro se utiliza esta librería para implementar las funciones de la lógica de negocio que gestionan la información espacial en el servidor.
- ❖ **Directorio *amd***: Este directorio define todos los módulos JavaScript desarrollados para la actividad que siguen el formato *AMD* (Asynchronous Module Definition) [20] necesario para la carga asíncrona de módulos que emplea Moodle a través del uso de *RequireJS* [21]. Está dividido en dos subcarpetas: en *src* se encuentran los ficheros sin minimizar<sup>3</sup> y en *build* los minimizados y listos para cachear en el servidor de Moodle. Para conseguir la versión minimizada de cada fichero, es necesario instalar el entorno JavaScript *nodejs* y su paquete de administración *npm* en el que se encuentra el automatizador de tareas *Grunt* [22]. Una vez instalado en el directorio principal de Moodle, sólo es necesario abrir una consola de *nodejs*, navegar hasta el directorio raíz de Moodle y ejecutar el comando `grunt uglify` que recorrerá todos los directorios *amd/src* en busca de módulos JavaScript que minimizar y colocar en cada carpeta *amd/build*.

En la caza del tesoro se han desarrollado los siguientes ficheros:

- **Fichero *edit.js***: se utiliza para crear la interfaz de edición de etapas y caminos e implementar los métodos que maneja. Se comunica con el servidor a través de *AJAX*, llamando a las funciones definidas en el fichero '*services.php*' descrito anteriormente que implementa el servicio web del módulo.
- **Fichero *play.js***: se utiliza para implementar los métodos de la interfaz de juego que emplea el usuario para resolver la caza del tesoro. Al igual que

---

<sup>3</sup> JavaScript es un lenguaje que se ejecuta en el cliente, por lo que es necesario descargar el fichero que lo contiene. Se considera que un fichero JavaScript está minimizado cuando ocupa el mínimo espacio posible, ahorrando ancho de banda, aunque provoca que sea ilegible por los desarrolladores.



en el anterior, la comunicación para las comprobaciones con el servidor se produce a través de AJAX.

- ***Fichero renewlock.js***: se utiliza para renovar el tiempo de bloqueo de edición de un usuario mientras se encuentra en la pantalla de edición de la caza, de un camino o una etapa. Como los anteriores, emplea AJAX para incrementar el valor en la base de datos a través del servicio web del módulo.
- ❖ **Directorio css**: Este directorio define todos los ficheros de estilo específicos de las librerías JavaScript utilizadas en el módulo.
- ❖ **Directorio js**: Este directorio contiene las librerías JavaScript externas, utilizadas como dependencias en los ficheros JavaScript del directorio *amd* descrito anteriormente para proporcionar determinadas funcionalidades al módulo. Como la carga de ficheros JavaScript está gestionada en Moodle a través de RequireJS, para incluir cada librería como una dependencia es necesario llamar a la función `require.config({})` en la cabecera de cada fichero *amd* antes de la definición del módulo. En esta función se debe escribir dentro de la propiedad *paths* el nombre junto con la situación en la estructura de ficheros de cada librería a incluir y, en el caso de que alguna no soporte el formato AMD, escribir dentro de la propiedad *shim* el nombre junto con el alias que recibirá la variable global que exportará todas las funciones de la librería cuando sea requerida en un módulo AMD.

El directorio está dividido en varios subdirectorios, cada uno del mismo nombre que la librería que almacena:

- **Directorio geocoder**: Este directorio contiene la librería JavaScript geocoder [23]. Se encarga de geocodificar las direcciones o las coordenadas que se pasan como parámetro utilizando distintos proveedores de servicio, como son Google Maps, Bing, OpenStreetMap, Mapquest o Yandex. En el módulo de la caza del tesoro se emplea dentro de la interfaz de edición y la interfaz de juego para permitir al usuario buscar la localización deseada y centrar el mapa en ella.
- **Directorio jquery-mobile**: Este directorio contiene el *framework* jQuery Mobile [24]. Se encarga de proporcionar el estilo y las herramientas para crear una interfaz amigable y adaptada a los dispositivos táctiles, como móviles y tablets. En el módulo de la caza del tesoro es utilizada para la interfaz de juego, ya que se presupone que se utilizarán *smartphones* por parte de los usuarios para resolver la actividad.
- **Directorio jquery-ui-touch-punch**: Este directorio contiene el fichero llamado *jquery-ui-touch-punch* [25]. Se encarga de modificar la librería JavaScript jQuery UI presente en Moodle para habilitar su interfaz de usuario para dispositivos táctiles, algo que por defecto no está disponible. Se utiliza en el módulo de la caza del tesoro en la interfaz de edición para permitir su uso en móviles y tablets.

- **Directorio *openlayers***: Este directorio contiene la librería JavaScript OpenLayers 3 [26]. Se encarga de proporcionar las herramientas necesarias para la creación de mapas interactivos con una arquitectura de capas. Permite utilizar capas base de tipo comercial (Bing, Yahoo, etc) o de licencia abierta (OpenStreetMap, MapQuest, etc) y desarrollar sobre ellos información geográfica de todo tipo, como añadir una capa vectorial de *features*. En el módulo de la caza del tesoro se utiliza para la creación del mapa disponible en la interfaz de edición y en la interfaz de juego, además de proporcionar las herramientas para la creación, edición y eliminación de las geometrías de las etapas.
- ❖ **Directorio *templates***: Este directorio contiene todas las plantillas en formato *mustache* [27] que siguen la recomendación de Moodle. Una plantilla *mustache* es una alternativa a escribir bloques de HTML en JavaScript o PHP mediante la concatenación de cadenas, ya que permite introducir variables que luego serán impresas con su valor durante el renderizado de la página. Tiene ciertas ventajas, como ver más claramente el resultado final sin introducir una lógica compleja o poder ser renderizadas desde JavaScript a través de AJAX. En la caza del tesoro se utiliza una única plantilla, localizada en el fichero '*play\_page.mustache*', para definir la estructura de la página que junto con el *framework* de jQuery Mobile explicado anteriormente da como resultado la interfaz de juego del módulo.
- ❖ **Fichero *LICENSE.txt***: Este fichero contiene la licencia de software del módulo para su distribución o modificación. Dado que es un módulo para la plataforma Moodle, debe contener una licencia GPL (GNU Public License) de versión 2 o 3. El módulo de la caza del tesoro está bajo los términos de la licencia GPL 3.
- ❖ **Fichero *README.md***: Este fichero contiene la descripción de lo que realiza el módulo y las instrucciones para comenzar a usarlo, como los pasos de instalación. Dado que el módulo de la caza del tesoro está publicado en Github se ha utilizado el formato más extendido, el Markdown, que transforma de forma sencilla un texto plano con sintaxis sencilla en un documento HTML válido.
- ❖ **Fichero *edit.php***: Fichero que define la pantalla de edición general de etapas y caminos del módulo. Contiene la comprobación de bloqueo de edición restringido a un único usuario y del permiso para administrar la actividad. Una vez superados llama a los módulos JavaScript *edit.js* y *renewlock.js* descritos anteriormente.
- ❖ **Fichero *editroad.php***: Fichero que define la pantalla de edición de un camino de la caza del tesoro. Al igual que las etapas, contiene la comprobación de bloqueo de edición restringido a un único usuario y de los permisos para editar o crear un camino. Una vez superados llama al módulo JavaScript *renewlock.js* descrito anteriormente y crea una instancia del formulario de un camino, pasándole los datos existentes si es una edición o manteniéndolos por defecto si es una creación.
- ❖ **Fichero *editroad\_form.php***: Fichero que define el formulario de configuración de los parámetros básicos de un camino. Este se muestra a los usuarios cuando se crea o se edita un camino, accediendo al fichero '*editroad.php*' definido anteriormente. Está compuesto por los siguientes campos:

- Nombre del camino
  - Selector del grupo o agrupación del curso que realiza este camino.
- ❖ **Fichero `editstage.php`:** Fichero que define la pantalla de edición de una etapa de la caza del tesoro. Contiene la comprobación de bloqueo de edición restringido a un único usuario y de los permisos para editar o crear una etapa. Una vez superados llama al módulo JavaScript *renewlock.js* descrito anteriormente y crea una instancia del formulario de una etapa, pasándole los datos existentes si es una edición o manteniéndolos por defecto si es una creación.
- ❖ **Fichero `editstage_form.php`:** Fichero que define el formulario de configuración de los parámetros básicos de una etapa. Este se muestra a los usuarios cuando se crea o se edita una etapa, accediendo al fichero *editriddle.php* definido anteriormente. Está compuesto por los siguientes campos:
- Nombre de la etapa
  - Descripción para localizar la siguiente etapa.
  - Selector de la actividad del curso a completar antes de mostrar la descripción de la etapa actual.
  - Selector para añadir una pregunta simple.
  - Texto de la pregunta. (Solo habilitado si en el selector se selecciona sí).
  - Texto de las respuestas. (Solo habilitado si en el selector se selecciona sí).
  - Casilla para seleccionar la respuesta correcta. (Solo habilitado si en el selector se selecciona sí).
- ❖ **Fichero `externallib.php`:** Como ya se ha mencionado, este fichero se ocupa de implementar todas las funciones definidas en el fichero '*services.php*' que constituyen el servicio web del módulo.
- ❖ **Fichero `grade.php`:** Este fichero define la página dentro del módulo a la que se redirigirá a un usuario procedente del calificador del curso.
- ❖ **Fichero `play.php`:** Este fichero define la pantalla de juego del módulo. Contiene la comprobación del permiso para jugar la actividad. Una vez superado, llama al módulo JavaScript *play.js* descrito anteriormente y a la función que renderiza la interfaz gráfica de la pantalla de juego.
- ❖ **Fichero `renderable.php`:** Este fichero contiene todas las clases necesarias para inicializar las funciones de renderizado disponibles en el fichero '*renderer.php*' que se explica a continuación. Todas ellas deben implementar la interfaz *renderable* propia de Moodle. Para el módulo de la caza del tesoro se incluyen cuatro clases:
- *treasurehunt\_user\_historical\_attempts*: Crea la clase con los atributos necesarios para luego generar el renderizado de la sección de intentos de un usuario o grupo que se muestra en la pantalla principal.
  - *treasurehunt\_info*: Crea la clase con los atributos necesarios para luego generar el renderizado de la sección de información de la actividad que se muestra en la pantalla principal

- *treasurehunt\_users\_progress*: Crea la clase con los atributos necesarios para luego generar el renderizado de la sección de progreso de todos los usuarios o grupos de la actividad, que se muestra en la pantalla principal.
- *treasurehunt\_play\_page*: Crea la clase con los atributos necesarios para luego generar el renderizado de la interfaz de juego de la actividad. A diferencia de las demás clases, esta necesita implementar la interfaz *templatable* que obliga a definir el método *export\_for\_template()* necesario para crear las variables que se pasarán a la plantilla del fichero '*play\_page.mustache*' explicado anteriormente.
- ❖ **Fichero *renderer.php***: Este fichero contiene la clase que extiende de *plugin\_renderer\_base* y cuyas funciones son las encargadas de devolver en una cadena el marcado HTML que se imprimirá en la pantalla apropiada. Como ya se ha comentado, requieren como parámetro cada una de las clases que renderizará.
- ❖ **Fichero *settings.php***: Este fichero define el formulario para el administrador de la plataforma con las opciones generales que se aplicarán por defecto a todas las nuevas instancias del módulo. Para la caza del tesoro permite administrar opciones que no dependen de la instancia, como el tiempo de renovación del bloqueo de edición de un usuario, o el tiempo de actualización del juego.
- ❖ **Fichero *thirdpartylibs.xml***: Este fichero contiene una lista en formato XML con el nombre, la localización relativa, la versión y el tipo de licencia de todas las librerías de terceros que incorpora el módulo.

#### 4.1.4. Diseño del modelo de datos

En el diseño de las tablas y columnas que conformarán los registros del módulo de la caza del tesoro, aparte de las pautas ya definidas por Moodle para la estructura, hay que tener en cuenta tres aspectos muy importantes:

- ❖ En primer lugar, la plataforma Moodle soporta distintas bases de datos sobre las que puede funcionar, y por tanto las sentencias para crear nuevas tablas y columnas en cada una de ellas pueden variar.
- ❖ En segundo lugar, todas las bases de datos soportadas por Moodle son de tipo relacional, por lo que implica que para un correcto diseño es necesario normalizar y atender a las relaciones entre tablas.
- ❖ En tercer lugar, existe un problema con el almacenamiento y gestión de los datos espaciales, es decir, las geometrías de las etapas y de los intentos de los usuarios.

Para el primer aspecto, Moodle proporciona una capa de abstracción que permite la compatibilidad entre todas las bases de datos soportadas y el código a desarrollar. Esa capa de abstracción es denominada XMLDB, y define de una manera neutral la creación y modificación de la arquitectura de tablas y columnas de la base de datos. La manera más fácil de utilizar XMLDB es a partir de la opción XMLDB editor, que permite crear el fichero '*install.xml*' ya mencionado en la arquitectura de ficheros del módulo sin esfuerzo ni errores en la estructura. Es accesible al entrar como administrador en la

plataforma y navegar hacia *Administración* → *Administración del sitio* → *Desarrollo* → *XMLDB editor*.

Para el segundo aspecto, es preciso conocer la definición y los pasos de la normalización. Normalizar se define como un 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 principales que transforman las relaciones no normales en normalizadas [28]:

- ❖ **Primera Forma Normal (1FN):** El primer paso consiste en evitar que en los atributos aparezcan varios valores: éstos tienen que ser elementales y únicos.
- ❖ **Segunda Forma Normal (2FN):** El segundo paso se centra en aislar en una tabla los datos que solo dependen de una parte de la clave, junto con dicha parte de la clave.
- ❖ **Tercera Forma Normal (3FN):** El tercer paso implica deshacerse de cualquier cosa de cada tabla que no dependa únicamente de la clave primaria.

También es preciso decidir las relaciones entre tablas. Dichas relaciones se clasifican en función del número de elementos que haya en cada lado de la relación. En concreto, pueden ser de los siguientes tipos:

- ❖ **Relación uno a uno:** Cada fila de una tabla tiene una, y solo una fila correspondiente en la tabla relacionada. Por este motivo, sería posible combinar las tablas relacionadas en una sola tabla compuesta por todas las columnas de las tablas relacionadas.
- ❖ **Relación uno a varios:** Vinculan una sola fila de una tabla con dos o más filas de otra tabla, mediante una relación entre la clave principal de la tabla base y la clave externa correspondiente en la tabla relacionada.
- ❖ **Relación varios a varios:** Las relaciones varios a varios no pueden expresarse como relaciones simples entre dos entidades. Para crearlas, es preciso que una tabla tenga una relación de varios a uno con dos tablas base.

Para el tercer aspecto, la primera solución pensada fue el uso de las columnas espaciales. Las bases de datos soportadas por Moodle disponen de distintas extensiones que incorporan los estándares OGC (Open Geospatial Consortium) para almacenar y realizar operaciones con la información espacial en columnas de tipo geométrico. Algunas, como MySQL desde la versión 5.0.16, MariaDB que deriva de esta última y Microsoft SQL Server desde la versión 2008, implementan su propia extensión por defecto. Sin embargo, la base de datos PostgreSQL necesita de la instalación aparte de la extensión Postgis, y Oracle Database necesita del uso de la opción Oracle Spatial and Graph. Sumado a esto, la capa de abstracción XMLDB proporcionada por Moodle para la interoperabilidad entre la plataforma y todas las bases de datos soportadas no reconoce la creación de columnas de tipo espacial. Debido a esto, se descartó esta primera aproximación en pro de la compatibilidad, aspecto muy promovido por Moodle. La segunda solución pasa por utilizar un campo de tipo texto dónde se guarden los campos espaciales en formato WKT

(Well Known Text). Esto soluciona los problemas del almacenamiento, sin embargo, esta solución deja en el aire el problema de las operaciones espaciales. Cuando un usuario envía un intento para comprobar si está dentro de la geometría de la etapa correspondiente debe realizarse una operación espacial que consiste en comprobar si un punto está dentro de un polígono. La opción más lógica es implementar dicho algoritmo en una función íntegramente en PHP a través de lo definido en [29].

Siguiendo estas pautas se ha conseguido generar un diseño de las tablas y campos del módulo válido para todas las bases de datos soportadas por Moodle y que evita los problemas de integridad y redundancia en los datos, además de permitir un máximo rendimiento a la plataforma cuando deba utilizar dichos datos. El resultado se muestra a continuación.

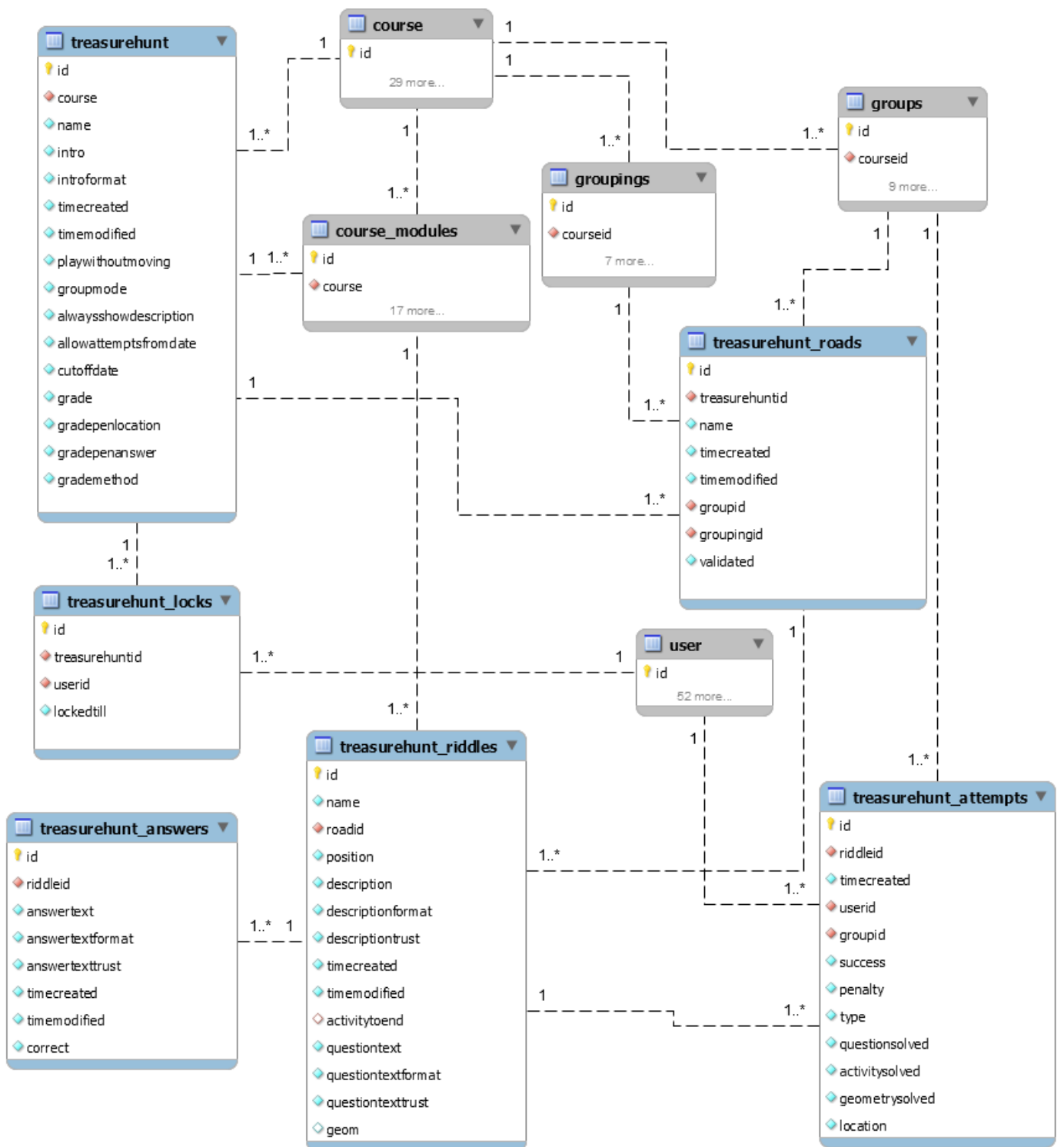


Figura 4.6 Esquema de la base de datos.

Como se puede observar, las tablas *course*, *course\_modules*, *groupings*, *groups* y *user* aparecen en un color diferente al resto. Esto se debe a que propiamente dicho no forman parte del módulo, son tablas pertenecientes a Moodle que han sido incluidas en el esquema porque existe una relación con las tablas que sí forman parte y aportan información de los datos requeridos de la plataforma para el correcto funcionamiento del módulo.

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

#### 4.1.4.1. Tabla treasurehunt

Esta tabla es la principal del módulo. En ella se define cada una de las instancias de la actividad presentes en todos los cursos de la plataforma. Almacena los valores principales de configuración.

treasurehunt				
Campo	Tipo	Nulo	Predet.	Comentario
id	bigint(10) [Primary Key]	No	-	Identificador de la instancia.
course	bigint (10) [Foreign Key]	No	0	Identificador del curso.
name	varchar (255)	No	-	Nombre de la instancia.
intro	longtext	No	0	Descripción de la instancia.
introformat	smallint (4)	No	0	Formato de la descripción.
timecreated	bigint (10)	No	0	Tiempo de creación de la instancia.
timemodified	bigint (10)	No	0	Tiempo de modificación de la instancia.
playwithoutmoving	tinyint (2)	No	0	Si es 1 indica que los estudiantes juegan la actividad sin moverse. Si es 0 tienen que ir a cada localización.
groupmode	tinyint (2)	No	0	Si es 1 indica que los estudiantes juegan la



				actividad en grupos. Si es 0 juegan individual.
alwaysshowdescription	tinyint (2)	No	0	Si es 1 indica que la descripción se mostrará siempre. Si es 0 solo se mostrará después de la fecha definida en el campo allowattemptsfromdate.
allowattemptsfromdate	bigint (10)	No	0	Fecha de inicio de la instancia.
cutoffdate	bigint (10)	No	0	Fecha de cierre de la instancia.
grade	bigint (10)	No	0	Nota máxima de la instancia.
gradepenlocation	decimal (10,5)	No	0.00000	Penalización por fallo en localización.
gradepenanswer	decimal (10,5)	No	0.00000	Penalización por fallo en respuesta.
grademethod	smallint (4)	No	1	Método de calificación.

Tabla 1: treasurehunt

#### 4.1.4.2. Tabla treasurehunt\_locks

Esta tabla reúne toda la información relativa a las sesiones de bloqueo de edición. Cada sesión de bloqueo está asociado a una única instancia de la caza del tesoro, pero una instancia del tesoro puede tener tantas sesiones de bloqueo de edición como quiera. De esta forma se cumple la relación uno a varios, y por ello esta misma tabla presenta un campo que almacena el identificador de la instancia.

treasurehunt_locks				
Campo	Tipo	Nulo	Predet.	Comentario
id	bigint (10) [Primary Key]	No	-	Identificador del bloqueo.
treasurehuntid	bigint (10) [Foreign Key]	No	0	Identificador de la instancia.
userid	bigint (10) [Foreign Key]	No	0	Identificador del usuario.

lockedtill	bigint (10)	No	0	Tiempo hasta que el bloqueo tiene validez.
------------	-------------	----	---	--

Tabla 2: treasurehunt\_locks

#### 4.1.4.3. Tabla treasurehunt\_roads

Esta tabla reúne toda la información relativa a los caminos. Cada camino está asociado a una única instancia de la caza del tesoro, sin embargo, una instancia puede tener tantos caminos como necesite. De esta forma se cumple la relación uno a varios, y por ello esta misma tabla presenta un campo que almacena el identificador de la instancia.

treasurehunt_roads				
Campo	Tipo	Nulo	Predet.	Comentario
id	bigint (10) [Primary Key]	No	-	Identificador del camino.
treasurehuntid	bigint (10) [Foreign Key]	No	0	Identificador de la instancia.
name	varchar (255)	No	-	Nombre del camino.
timecreated	bigint (10)	No	0	Tiempo de creación del camino.
timemodified	bigint (10)	No	0	Tiempo de modificación del camino.
groupid	bigint (10) [Foreign Key]	No	0	Tiempo de creación de la respuesta.
groupingid	bigint (10) [Foreign Key]	No	0	Tiempo de modificación de la respuesta.
validated	tinyint (2)	No	0	Si es 1 indica que el camino está validado.

Tabla 3: treasurehunt\_roads

#### 4.1.4.4. Tabla treasurehunt\_stages

Esta tabla reúne toda la información relativa a las etapas. Cada etapa está asociada a un único camino, sin embargo, un camino puede tener tantas etapas como necesite. De esta forma se cumple la relación uno a varios, y por ello esta misma tabla presenta un campo que almacena el identificador del camino.

treasurehunt_stages				
Campo	Tipo	Nulo	Predet.	Comentario

id	bigint (10) [Primary Key]	No	-	Identificador de la etapa.
name	varchar (255)	No	-	Nombre de la etapa.
roadid	bigint (10) [Foreign Key]	No	0	Identificador del camino.
position	bigint (10)	No	0	Orden de la etapa dentro del camino.
cluetext	longtext	No	-	Descripción de la pista.
clueformat	tinyint (2)	No	0	Formato de la descripción de la pista.
clustrust	tinyint (2)	No	0	Si es 1 indica que el texto de la descripción de la pista es confiable.
timecreated	bigint (10)	No	0	Tiempo de creación de la etapa.
timemodified	bigint (10)	No	0	Tiempo de modificación de la etapa.
playstagerwithoutmoving	tinyint (2)	No	0	Si es 1 indica que los estudiantes juegan la etapa sin moverse. Si es 0 el método de juego depende de la configuración general.
activitytoend	bigint (10) [Foreign Key]	No	0	Identificador de la actividad a finalizar.
questiontext	longtext	No	-	Descripción de la pregunta.
questiontextformat	tinyint (2)	No	0	Formato de la descripción de la pregunta.
questiontexttrust	tinyint (2)	No	0	Si es 1 indica que el texto de la descripción de la pregunta es confiable.

geom	longtext	Sí	NULL	Multipolígono de localización de la etapa en formato WKT.
------	----------	----	------	---

Tabla 4: *treasurehunt\_stages*

#### 4.1.4.5. Tabla *treasurehunt\_answers*

Esta tabla reúne toda la información relativa a las respuestas. Cada respuesta está asociada a una única etapa cuyo campo pregunta no esté vacío, sin embargo, una etapa con pregunta puede tener tantas respuestas como necesite. De esta forma se cumple la relación uno a varios, y por ello esta misma tabla presenta un campo que almacena el identificador de la etapa.

<b>treasurehunt_answers</b>				
Campo	Tipo	Nulo	Predet.	Comentario
id	bigint (10) [Primary Key]	No	-	Identificador de la respuesta.
stageid	bigint (10) [Foreign Key]	No	0	Identificador de la etapa.
answertext	longtext	No	-	Texto de la respuesta.
answertextformat	tinyint (2)	No	0	Formato del texto de la respuesta.
answertexttrust	tinyint (2)	No	0	Si es 1 indica que el texto de la respuesta es confiable.
timecreated	bigint (10)	No	0	Tiempo de creación de la respuesta.
timemodified	bigint (10)	No	0	Tiempo de modificación de la respuesta.
correct	tinyint (2)	No	0	Si es 1 indica que la respuesta es correcta.

Tabla 5: *treasurehunt\_answers*

#### 4.1.4.6. Tabla *treasurehunt\_attempts*

Esta tabla reúne toda la información relativa a los intentos de los usuarios. Cada intento está asociado a una única etapa, sin embargo, una etapa puede tener tantos intentos como necesite para resolverse. De esta forma se cumple la relación uno a varios, y por ello esta misma tabla presenta un campo que almacena el identificador de la etapa.

El campo *groupid* almacena el identificador del grupo si el intento se realiza en una instancia que tiene asignado el modo grupo, sino significa que el intento es realizado a nivel individual y el valor de este campo sería 0.

El campo *userid* siempre almacena el identificador del usuario que realiza el intento, a pesar de que el modo de juego sea en grupo, ya que así aporta información sobre qué usuario dentro del grupo realiza cada intento y de esta manera poder informar de actualizaciones al resto de usuarios del grupo o al profesor que revise el historial de intentos.

El campo *type* indica el tipo de intento al que hace referencia, pudiendo tomar los valores de “*location*” cuando el intento pretende resolver la localización de una etapa, “*question*” cuando el intento pretende resolver la pregunta asociada a una etapa o “*activity*” cuando el intento pretende resolver la actividad a completar asociada a la etapa.

treasurehunt_attempts				
Campo	Tipo	Nul o	Predet.	Comentario
id	bigint (10) [Primary Key]	No	-	Identificador de la respuesta.
stageid	bigint (10) [Foreign Key]	No	0	Identificador de la etapa.
timecreated	bigint (10)	No	0	Tiempo de creación del intento.
userid	bigint (10) [Foreign Key]	No	0	Identificador del usuario.
groupid	bigint (10) [Foreign Key]	No	0	Identificador del grupo.
success	tinyint (2)	No	0	Si es 1 la etapa está resuelta.
penalty	tinyint (2)	No	0	Si es 1 hay penalización en el intento.
type	varchar (20)	No	location	Indica el tipo de intento.
questionsolved	tinyint (2)	No	0	Si es 1 la pregunta asociada a la etapa está resuelta.
activitiesolved	tinyint (2)	No	0	Si es 1 la actividad asociada a la etapa está resuelta.
geometrysolved	tinyint (2)	No	0	Si es 1 la geometría asociada a la etapa está resuelta.
location	longtext	No	-	Punto de localización del intento en formato WKT.

Tabla 6: treasurehunt\_attempts

## 4.2. Aspectos funcionales

Este apartado detalla las características que ofrece el módulo “La caza del tesoro” y los pasos dados para implementarlas con el fin de cumplir con los objetivos fijados. Cuando se pensó inicialmente en la estructura que debía seguir la aplicación a nivel funcional, surgieron dos grandes bloques: Por un lado, era ineludible que el profesor pudiese disponer de las herramientas necesarias para configurar y dar forma a la actividad. De ello surgió el bloque de administración de la actividad, que incluye el formulario general de configuración y el entorno para crear y editar los caminos y etapas de los que está formada la caza. Por otro lado, se debía dotar a la aplicación de su principal función, la de permitir a los alumnos jugar la actividad. Este bloque se describe en el apartado “Jugar”.

A continuación, se detalla la implementación de cada uno de los bloques mencionados, además de incluir un apartado a mayores para aquellas funcionalidades que no pueden incluirse en esas categorías, pero que aportan grandes ventajas a la aplicación y amplían su compatibilidad con la plataforma Moodle.

### 4.2.1. Administración de la actividad

Este bloque abarca la implementación de todas las funcionalidades creadas para que el rol de profesor pueda crear y configurar una caza del tesoro para sus alumnos. La administración de una actividad puede verse en dos grandes grupos: La configuración general, y el entorno de creación y edición.

#### 4.2.1.1. Configuración general

Esta funcionalidad está orientada al uso exclusivo por parte del profesor con permisos de edición del curso, aunque es posible acceder a ella siempre que se disponga de los permisos de gestión de actividades en el contexto del curso. Su finalidad es definir el formulario para alterar los parámetros generales del juego. Este se muestra a los profesores cuando se crea o se edita una instancia de la actividad.

#### **Parámetros de configuración**

Atendiendo a los requisitos de configuración del juego del apartado 3.2.2 se ha decidido crear un formulario con diferentes pestañas, las cuales contienen los parámetros de configuración de la actividad, dependiendo del servicio que facilitan dentro del módulo. Para ello se ha considerado en todo momento el uso de la API de formularios de la que dispone Moodle y que facilita en gran medida la labor del desarrollador.

Para empezar, se define una pestaña “general”, que debe contener los campos más universales de configuración de la actividad. Estos son:

- ❖ **Nombre de la caza del tesoro:** Un campo de tipo texto para introducir el nombre que se mostrará en la pantalla de cursos de Moodle. Utilizando las funciones de formularios de Moodle se define el campo como obligatorio, y se añade una norma para que el campo no supere cierto número de caracteres.

- ❖ **Descripción:** Un campo de tipo *editor* con la descripción de la actividad. Este tipo de campo desarrollado por Moodle permiten la subida de contenido multimedia, incluyendo texto formateado.
- ❖ **Jugar sin desplazarse:** Un campo de tipo *checkbox* que permite al profesor controlar el método de juego. Uno de los requisitos de configuración fijado era permitir al profesor seleccionar si la actividad se realizaba acudiendo presencialmente a las geometrías de cada etapa o bien mediante una marca sobre el mapa que indicara la localización que el usuario deseaba enviar.

Como existen algunos campos como el anterior que pueden ser confusos de entender solo con el título que ofrecen, Moodle proporciona la posibilidad de incluir unos iconos de ayuda para describir más minuciosamente de lo que se ocupan. Tanto en el campo anterior como de aquí en adelante, se usarán en todos los campos que puedan necesitar una mínima explicación o ejemplo.

Otro de los requisitos de configuración definía que el profesor tendría la capacidad de escoger la fecha de inicio de la actividad y fecha de cierre, a partir de la cual no se podrían enviar más intentos. Para dar cabida a ello se define una pestaña de “Disponibilidad” que contiene los campos:

- ❖ **Permitir intentos desde:** Un campo de tipo *date\_time\_selector* para escoger la fecha de inicio de la actividad. Este tipo de campo desarrollado por Moodle posibilita la selección de una fecha y hora y la capacidad de habilitarla o deshabilitarla. De este modo, si está habilitada no se permitirá a los alumnos jugar antes de la fecha.
- ❖ **Fecha límite:** Un campo de tipo *date\_time\_selector* para escoger la fecha de cierre de la actividad. De este modo, si está habilitada no se permitirá a los alumnos seguir jugando a partir de la fecha.
- ❖ **Mostrar siempre la descripción:** Un campo de tipo *checkbox* que permita deshabilitar la descripción antes mencionada en caso de que esté habilitada una fecha de inicio.

También se hace necesario una pestaña dónde se gestionen las configuraciones relativas a los grupos dentro de la actividad. Moodle incorpora su propio sistema de creación de grupos y agrupaciones, por lo que esta pestaña no debe ocuparse de ello. La pestaña incluye un único campo:

- ❖ **Juego en grupos:** Un campo de tipo *checkbox* que permite al profesor controlar el modo de juego. En los requisitos de configuración se definió que el profesor podría escoger si la actividad se realizaba en modo grupo o individual.

Otra de las pestañas que se hacen necesarias es la de “Calificación”. Para ello, Moodle proporciona una función que se encarga de crear los campos más significativos, como el tipo de calificación, la puntuación máxima, la categoría y la calificación necesaria para aprobar la actividad. Sin embargo, siguiendo los requisitos de configuración deben existir más campos de los proporcionados. Estos son los añadidos para la calificación:

- ❖ **Método de calificación:** Un campo de tipo selector que permita al profesor elegir el método de calificación deseado. En los requisitos de configuración se definió que el profesor podría escoger entre diferentes métodos de calificación. Para ello, se han definido tres tipos distintos, atendiendo a las posibilidades que proporciona un juego como la caza del tesoro. El primer tipo definido es “puntuación por etapas”. En él, cada jugador o grupo puntúa de forma proporcional al número de etapas resueltas, siendo el 100% de la calificación máxima cuando se ha completado el camino y 0 cuando no se ha resuelto ninguna etapa. El segundo tipo es “puntuación por tiempo”. En él la calificación se calcula interpolando el tiempo de finalización, siendo el 50% de la calificación máxima el peor tiempo de finalización y el 100% el mejor. Los jugadores o grupos que no terminaron la caza reciben una calificación por debajo del 50% calculado simplemente por el número de etapas resueltas. El último tipo es “puntuación por posición”. En él la puntuación se calcula interpolando la posición en el ranking, siendo el 100% de la calificación máxima para el primer jugador o grupo en finalizar y 50% para el último jugador. Los jugadores que no terminaron la caza reciben una calificación por debajo del 50% calculado simplemente por el número de etapas resueltas.
- ❖ **Penalización por fallo en localización:** Un campo de tipo *float* que permite al profesor elegir el porcentaje de la calificación total del alumno que se restará por cada fallo en una localización.
- ❖ **Penalización por fallo en respuesta:** Un campo de tipo *float* que permite al profesor elegir el porcentaje de la calificación total del alumno que se restará por cada fallo en una respuesta.

Además, estos tres campos adicionales se han configurado para que solo aparezcan habilitados si el tipo de calificación definido es “puntuación”, de otro modo no tendría sentido utilizarlos.

Moodle también proporciona la posibilidad de añadir otras pestañas comunes a todos los módulos, como “Ajustes comunes del módulo” dónde se configura la visibilidad de la actividad, “Restricciones de acceso” que permiten añadir restricciones que deben cumplirse para entrar en la actividad o “Finalización de actividad” que permite especificar cuándo se considera la actividad como finalizada.

Al final del formulario, se generan los botones de guardar cambios y regresar al curso, guardar cambios y mostrar (redirige a la pantalla principal de la actividad), y cancelar, perdiendo cualquier dato introducido y redirigiendo a la pantalla principal de la actividad.

Como es natural, antes de permitir al usuario guardar la configuración, se deben realizar comprobaciones de los campos para que no existan incongruencias, tales como que la fecha límite de la actividad sea menor que la fecha de inicio, o que los campos de penalización, definidos para actuar como porcentaje, tenga un valor superior al 100%.



#### 4.2.1.2. Entorno de creación y edición

Esta funcionalidad está orientada al uso exclusivo por parte del profesor con permisos de edición del curso, aunque es posible acceder a ella siempre que se disponga de los permisos de gestión de la caza del tesoro en el contexto de la actividad. Su finalidad es proporcionar una interfaz al profesor que le permita crear todos los caminos y etapas que formarán el juego de la caza del tesoro.

##### **Decisiones de diseño**

Antes de comenzar a desarrollar el código para la implementación del entorno de creación y edición de etapas y caminos, se han tenido en cuenta diversos aspectos:

- ❖ Es primordial que la interfaz resultante sea atractiva, intuitiva, rápida e interactiva, proporcionando una buena experiencia de uso para el usuario final. Aunque en el contexto de la aplicación ese usuario sea el profesor, no se debe dar por hecho que entiende el pleno funcionamiento de la actividad, por ello se le debe ir guiando en el proceso de creación y edición de los caminos y las etapas.
- ❖ Las funcionalidades individuales, como crear un camino, deben ser seguras pero accesibles individualmente. No se debe especular con la posibilidad de que el rol del profesor sea el único que pueda editar y crear los caminos y las etapas. En muchos cursos, el profesor principal delega trabajos específicos en otros roles, sin querer permitir que realicen más allá de la función que se les ha encomendado. Aquí es donde toma importancia el sistema de permisos de Moodle, pudiendo asignar permisos específicos para cada funcionalidad concreta.
- ❖ Se debe garantizar la integridad de los datos generados. La integridad se ve comprometida si dos o más profesores pueden acceder a la interfaz de edición al mismo tiempo, estropeándose los cambios realizados mutuamente y corrompiendo los datos de caminos y etapas.

Atendiendo a estos aspectos y a los requisitos de trabajo impuestos en el apartado 3.2, la primera cuestión que se plantea es qué tecnologías usar para crear la interfaz de edición. Moodle recomienda usar principalmente PHP con el objetivo de garantizar la accesibilidad. Sin embargo, la experiencia de usuario producida por una interfaz enteramente diseñada en PHP, donde cada nueva acción provoca que la página se recargue, se ve resentida por los tiempos de carga, afectando a un aspecto muy importante en interfaces de aplicación como es la interactividad. Aquí es donde entran en juego el uso conjunto de tecnologías como JavaScript y AJAX (Asynchronous JavaScript and XML). Su uso combinado permite realizar consultas al servidor sin necesidad de recargar la página, lo que proporciona el poder de modificar la interfaz mediante JavaScript al recibir la respuesta, mejorando la interactividad, velocidad y usabilidad. Por ello, parece una tecnología atractiva sobre la que construir el funcionamiento de la interfaz de edición.

Moodle, a partir de su versión 2.9, proporciona respuesta a peticiones AJAX empleando el formato JSON a través de los servicios web. Para implementarlo en la caza del tesoro, es necesario crear un servicio web específico del módulo, definir las funciones que soporta y habilitar en cada una la posibilidad de consultas AJAX. Las peticiones AJAX

deben hacerse desde los ficheros JavaScript propios del módulo de la caza del tesoro. Estos ficheros deben seguir el nuevo formato AMD (Asynchronous Module Definition) que permite crear pequeñas piezas de código con funciones asociadas y encapsularlas como un “módulo” JavaScript. Estos módulos permiten llamadas a otros módulos, cargándolos asíncronamente cuando son necesarios, de modo que se puedan emplear funciones de los módulos JavaScript del núcleo de Moodle, como la necesaria para crear una consulta AJAX, sin relativo esfuerzo para el programador.

Sin embargo, no todas las funcionalidades de edición se crearán usando el servicio AJAX, ya que los formularios de Moodle que se emplearán en la edición de las características de caminos y etapas, y que facilitan en gran medida la labor del desarrollador, no soportan esta tecnología.

La segunda cuestión que se plantea en el diseño de la interfaz, viene a la hora de soportar la información espacial, es decir, la creación y modificación de geometrías que se asignarán a cada etapa. Para ello, existen librerías específicas JavaScript como Openlayers 3 [22], que se encargan de proporcionar las herramientas necesarias para la creación y edición de mapas interactivos siguiendo una arquitectura de capas. En cada capa, es posible cargar una colección de “*features*”, mostrarlas con la configuración de estilo deseada, y crear nuevas o modificarlas utilizando las funciones de las que dispone la librería. Como ya se explicó, una *feature* en el argot de los sistemas de información geográfica se define como un objeto que tiene atributos espaciales (la geometría que utiliza, ya sean puntos, líneas, polígonos o multipolígonos con su ubicación en el plano) y atributos no espaciales que dicen algo sobre esa geometría, como su nombre, descripción, etc.

Sin embargo, el requisito de trabajo que obliga a soportar varias geometrías por cada etapa, supone un bache técnico. El problema radica en que la librería Openlayers 3 solo permite crear geometrías de tipo polígono, mientras que la forma más comprensible de almacenar la geometría total de una etapa es mediante un multipolígono. Para solucionarlo y a la vez conseguir la máxima eficiencia cuando se tengan que guardar los datos editados en el servidor, se debe crear un estado de abstracción conformado por tres contenedores de *features*:

- ❖ Un primer contenedor dónde se almacenen todas las *features* que representan en su conjunto el estado de las etapas guardadas en la base de datos. En este, cada *feature* constituye una etapa, por lo que su geometría es de tipo multipolígono.
- ❖ Un segundo contenedor dónde se almacenen todas las *features* representadas en el mapa, compuestas de un único polígono y que almacenan en sus atributos el identificador de la etapa a la que pertenece su geometría.
- ❖ Un tercer contenedor igual que el primero, pero donde solo se almacenan aquellas *features* que representan etapas editadas respecto a la información de la base de datos, es decir, aquellas que han cambiado la posición que ocupaban en el camino, o que su geometría total se ha visto modificada de alguna manera.

De esta forma será posible manipular los polígonos visibles en el mapa, ya sea editándolos, eliminándolos o creando nuevos, y al terminar con cada interacción guardar los cambios producidos en el tercer contenedor.

Hasta ahora se ha hablado del manejo de los datos espaciales en el cliente, pero estos deben ser enviados de alguna forma al servidor para su procesamiento y almacenamiento. Por lo tanto, se hace necesario un método de encapsulación de la información geográfica para el envío y la recepción de datos espaciales entre el cliente y el servidor. Uno de los formatos más eficientes es GeoJSON [18], que posibilita la transferencia de grupos de *features*. Este formato es plenamente soportado por Openlayers para la creación de capas de *features*, a la vez que permite la transformación inversa. Sin embargo, se hace necesario en el servidor una librería que soporte el formato GeoJSON y permita crear y gestionar las *features* para su inserción y recuperación de la base de datos. Una librería PHP que incluye todos los requisitos necesarios indicados y que recibe el mismo nombre que el formato es GeoJSON. Esta librería forma parte del framework MapFish [17], utilizado para crear aplicaciones de mapas web.

Así, una comunicación típica entre cliente y servidor como es la de solicitar todos los caminos y etapas pertenecientes a una instancia, vendría dada por una petición AJAX que encapsula su contenido en formato JSON, como la de la Figura 4.7. En ella se indica un objeto con el nombre del método a llamar en el servicio web, en este caso la función “*fetch\_treasurehunt*”, y los argumentos requeridos, concretamente la instancia 2. En el servidor, la función del servicio web realiza todas las comprobaciones de permisos, y si es todo correcto procesa la información solicitada y devuelve una respuesta en formato JSON, como la de la Figura 4.8. En este caso, devuelve un objeto *treasurehunt* con la lista de todos los caminos. Dentro de cada camino se definen sus propiedades, como su identificador, el nombre, si está bloqueado y un objeto que contiene la lista de todas las etapas pertenecientes al camino siguiendo el formato GeoJSON. También devuelve un objeto para explicar el estado de la consulta. Para este caso en concreto, la caza del tesoro se ha cargado con éxito.

```
▼ 0: Object
  index: 0
  methodname: "mod_treasurehunt_fetch_treasurehunt"
  ▼ args: Object
    treasurehuntid: "3"
```

Figura 4.7 Petición en formato JSON al servidor web del módulo

```

  ▼ treasurehunt: Object
    ▼ roads: Object
      ▼ 0: Object
        id: 7
        name: "Camino 1"
        blocked: false
        ▼ stages: Object
          type: "FeatureCollection"
          ▼ features: Array
            ▶ 0: Object
            ▼ 1: Object
              type: "Feature"
              id: 14
              ▼ geometry: Object
                type: "MultiPolygon"
                ▶ coordinates: Array
              ▼ properties: Object
                roadid: 7
                stageposition: 1
                name: "Etap 1"
                treasurehuntid: 3
                clue: "<p>Esta es la pista 1</p>"
            ▼ status: Object
              code: 0
              msg: "La caza del tesoro se ha cargado con éxito"

```

Figura 4.8 Respuesta en formato JSON al servidor web del módulo

La tercera cuestión que se presenta en el diseño de la interfaz nace de la necesidad de garantizar la integridad de los datos, es decir, cómo conseguir que los profesores no puedan editar al mismo tiempo. Tras mucho pensar y estudiar una solución se llegó a la conclusión de que la mejor manera de prevenir la edición simultánea era creando un bloqueo de edición. Su funcionamiento es sencillo: Cuando un usuario accede a la edición de una instancia se comprueba si ya existe otro usuario que esté bloqueando el servicio. Si existe, se muestra un mensaje de error notificando el usuario que está haciendo uso de ella. Si no existe, se crea un bloqueo de edición durante X segundos (definidos en las propiedades de administración de la actividad) y este se va renovando antes de que finalice a través de peticiones desde el cliente al servidor mientras el usuario permanezca en la página. Una vez que el usuario abandona la edición, al acabar el tiempo de bloqueo la página vuelve a estar accesible para otros usuarios. Para implementarlo en la caza del tesoro se han creado distintas funciones:

- ❖ Una función que comprueba si ya hay alguien editando la caza del tesoro.
- ❖ Otra función que crea o renueva un registro en la base de datos, con el identificador del usuario que bloquea la edición, la instancia bloqueada y el tiempo en el que expirará ese bloqueo. Además, devuelve como “ticket” el identificador de dicho registro y elimina cualquier registro de otro usuario para esa instancia.
- ❖ Otra función que comprueba, antes de llamar a la función de renovar el tiempo de bloqueo, si el “ticket” sigue existiendo en la base de datos. Es necesaria para casos extraños. Por ejemplo, si un usuario “A” edita la caza, no guarda los últimos cambios realizados, suspende su equipo sin cerrar la ventana de edición y el tiempo de bloqueo se agota. Llega otro usuario “B”, edita la caza, guarda y cierra. Al restaurar el equipo el usuario “A” se encontrará con que la caza no está

bloqueada pero su ticket ya no será válido, por lo que no le dejará guardar sus cambios al no tener la última información actualizada de los caminos y las etapas de la instancia.

También se ha necesitado de la implementación de un módulo JavaScript que, utilizando AJAX para comunicarse con el servicio web en el servidor, llame a la función de comprobar y renovar el “ticket” antes de que se agote el tiempo de bloqueo de edición.

Para que el bloqueo sea útil, debe existir una comprobación de “tickets” en cada una de las funciones disponibles en el servicio web que implementan alguna de las funcionalidades de la interfaz de edición. De esta forma, si el “ticket” no es válido no se realizarán los cambios que intente realizar el usuario, y se le informará de la pérdida del bloqueo de edición de la página.

Por último, para la creación de una interfaz gráfica se debe tener en cuenta las partes que la componen. En primer lugar, debe existir un panel de selección del camino a configurar. En cada camino, debe aparecer una serie de botones que permitan borrarlo o modificar sus parámetros generales. En segundo lugar, debe existir un panel de selección de las etapas existentes en el camino elegido. En cada etapa, deben aparecer como mínimos los mismos botones que en cada camino, para permitir borrarlas o modificar sus parámetros generales. En tercer lugar, debe existir un mapa dónde se muestren representadas todas las geometrías de cada etapa en el camino. Finalmente, una barra de herramientas para la creación de nuevos caminos y etapas, y para la gestión de las geometrías asociadas a cada etapa. Para realizar todo esto, es recomendable apoyarse en librerías que facilitan la labor del desarrollador. Moodle incorpora como módulos JavaScript la librería jQuery, que proporciona la manipulación del árbol DOM y el manejo de eventos de un documento HTML, y la librería jQuery UI, que provee interacciones, controles y efectos visuales que simplifican la creación de una interfaz gráfica de aplicación. Sin embargo, esta última por defecto no gestiona las interacciones en dispositivos táctiles. Para habilitarlo, es necesario añadir el script jQuery UI Touch Punch [25]. Ambas librerías se emplearán en el módulo JavaScript que se utilizará para la creación de la interfaz de edición, y que incorporará las llamadas AJAX a las funciones necesarias del servicio web de la caza del tesoro.

### **Funcionalidades incorporadas en la interfaz**

Una vez definidas las tecnologías a emplear, la comunicación con el servidor, y las partes de las que consta la interfaz gráfica, es necesario concretar las funcionalidades que otorga el entorno de edición. Como se explicó, la interfaz solo se comunica con el servidor a través de AJAX para aquellas funcionalidades que no requieran del uso de un formulario. De esta manera, la interfaz se centrará en la creación, edición y borrado de las geometrías pertenecientes a cada etapa, al orden de las etapas dentro de un camino, al borrado de etapas y caminos, y a proporcionar los enlaces que comunican con los formularios de creación y edición de estos últimos.

A continuación, se detalla cada funcionalidad junto con los pasos dados para implementarla:

- ❖ **Seleccionar camino.** Esta funcionalidad se ha desarrollado utilizando CSS y los estilos y animaciones disponibles en Openlayers 3. Al seleccionar un camino, la

página mueve el *scroll* centrando la interfaz y cambia el color a azul de su contenedor, permitiendo resaltarlo sobre el resto de caminos si los hubiese. También desmarca cualquier etapa seleccionada y cambia el color en el mapa de todas las geometrías de las etapas del camino a morado, encuadrando el mapa en torno a ellas. Si el camino seleccionado no tiene dos o más etapas, muestra un aviso indicándolo y resalta el botón de crear etapa del panel de herramientas. Si cumpliendo el requisito anterior, una etapa no dispone de geometría asociada, muestra un aviso indicándolo. Al entrar en la interfaz, se selecciona automáticamente el camino que esté como parámetro URL, si éste es válido, o bien el primer camino de la instancia, con el fin de que siempre haya un camino seleccionado. Si no existe ningún camino, se redirige al usuario al formulario de creación de un nuevo camino para la instancia. Para dar acceso al usuario a esta funcionalidad, se ha creado en la interfaz un panel de caminos.

- ❖ **Seleccionar etapa.** Esta funcionalidad se ha desarrollado utilizando CSS y los estilos y animaciones disponibles en Openlayers 3. Al seleccionar una etapa cambia el color a azul de su contenedor, permitiendo resaltarla sobre el resto de etapas. También cambia el color de morado a amarillo de las geometrías asociadas a la etapa y centra el mapa entorno a ellas. Si la etapa seleccionada no dispone de geometría asociada, se resalta el botón de añadir geometría del panel de herramientas. Empleando esta funcionalidad se habilita la utilización de las funcionalidades de añadir, editar y borrar geometría. Para dar acceso al usuario a esta funcionalidad, se ha creado en la interfaz un panel de etapas en el que se visualizan todas las que corresponden al camino seleccionado, y se esconden el resto.
- ❖ **Añadir geometría.** Esta funcionalidad se ha desarrollado utilizando la interacción de dibujar de la librería Openlayers 3. Permite añadir polígonos a una etapa pulsando *clicks* sobre el mapa. Cada *click* supone un vértice de la geometría. Para finalizar la creación es necesario hacer doble *click*, estableciéndose el vértice final. Además, la etapa a la que se añade la nueva geometría pasa al contenedor de cambios sin guardar. También se ha implementado un método que cancela el dibujo en curso usando la tecla “Esc”. Para dar acceso al usuario a esta funcionalidad, se ha colocado en el panel de herramientas de la interfaz un botón con el símbolo “+”. Este botón solo se habilita si previamente se ha seleccionado una etapa a la que añadir el nuevo polígono.
- ❖ **Editar geometría.** Esta funcionalidad se ha desarrollado utilizando los estilos y las interacciones de modificar y seleccionar de la librería Openlayers 3. Permite modificar únicamente los polígonos pertenecientes a una etapa, seleccionándolos previamente y arrastrando sus vértices a la nueva posición deseada. Cuando se selecciona un polígono, este cambia a color azul para diferenciarlo del resto. Al terminar la interacción, la etapa modificada pasa al contenedor de cambios sin guardar. Se ha puesto especial importancia en solo permitir marcar para modificar aquellas geometrías que corresponden a la etapa seleccionada. Para dar acceso al usuario a esta funcionalidad, se ha colocado en el panel de herramientas de la interfaz un botón con el símbolo característico de un lapicero. Este botón solo se habilita si previamente se ha seleccionado una etapa.

- ❖ **Eliminar geometría.** Esta funcionalidad se ha desarrollado utilizando la interacción de seleccionar de la librería Openlayers 3. Permite eliminar los polígonos seleccionados mediante la opción de editar geometría. Antes de proceder a eliminar se crea un cuadro de diálogo para que el usuario confirme su intención. Al terminar la interacción, la etapa modificada pasa al contenedor de cambios sin guardar. Para dar acceso al usuario a esta funcionalidad, se ha colocado en el panel de herramientas de la interfaz un botón con el símbolo característico de un cubo de basura. Este botón solo se habilita si previamente se ha seleccionado una geometría.
- ❖ **Cambiar el orden de las etapas.** Esta funcionalidad se ha desarrollado empleando la interacción “*Sortable*” de la librería jQuery UI. Permite mover cada etapa, desde la zona habilitada para ello, arriba o abajo para cambiar su orden dentro de un camino. Al terminar la interacción, actualiza mediante JavaScript el número con la posición que ocupan tanto cada etapa implicada como las geometrías asociadas representadas en el mapa. También añade al contenedor de cambios sin guardar las etapas cuya posición se ha visto afectada. Esta funcionalidad solo está disponible si el camino no está bloqueado, es decir, si ningún usuario/grupo ha realizado un intento sobre ese camino.
- ❖ **Control total del mapa:** La librería Openlayers 3 garantiza un control total sobre la visualización del mapa para todo tipo de dispositivos, incluyendo móviles. Permite hacer zoom mediante los botones habilitados, además de con la rueda del ratón o a través de “pellizcos” en dispositivos táctiles. Además, se ha desarrollado un botón incrustado en el mapa que emplea la interacción “*Resizable*” de la librería jQuery UI para controlar el tamaño horizontal del panel de etapas respecto del mapa. Este botón desaparece si el panel de etapas no contiene ninguna etapa, ampliando el mapa al máximo permitido.
- ❖ **Guardar cambios:** Esta funcionalidad se ha desarrollado mediante la herramienta de conversión de formato a GeoJSON de la librería Openlayers 3. Permite guardar los cambios realizados en las posiciones y geometrías de las etapas. Para ello, convierte el contenedor con las *features* de las etapas modificadas en formato GeoJSON y las envía al servidor mediante AJAX, dónde los cambios son procesados. Si el usuario dispone del permiso de “*Editar etapa*” y el “*ticket*” del bloqueo de edición sigue siendo válido, los cambios se guardan en la base de datos y la petición AJAX responde satisfactoriamente. Si no, la petición AJAX responderá con un error y no se guardará ningún cambio. Una vez que la petición AJAX responde satisfactoriamente, el contenedor de cambios aplica las modificaciones sobre el contenedor que almacena el actual estado de la base de datos y luego se vacía. Para dar acceso al usuario a esta funcionalidad, se ha colocado en el panel de herramientas de la interfaz un botón con el símbolo característico de un disquete de ordenador. Este botón solo se habilita si el contenedor de cambios no está vacío.
- ❖ **Buscar localización.** Esta funcionalidad se ha desarrollado mediante el uso de las animaciones de Openlayers, la utilización del *widget* de autocompletado de jQuery UI y de la librería geocoder [23]. Permite insertar en el buscador una

dirección o unas coordenadas, mostrando un desplegable con los resultados más aproximados. Para ello, se realiza a través de la librería geocoder [23] una consulta al servicio de nombres de *Open Street Map*, el cual devuelve las respuestas. Al seleccionar uno de los resultados encuadra el mapa en la ubicación deseada. Para dar acceso al usuario a esta funcionalidad, se ha colocado en el panel de herramientas de la interfaz un campo de texto que incluye una animación de carga escondida para mostrar mientras se realiza la petición.

- ❖ **Mostrar información de la etapa.** Esta funcionalidad se ha desarrollado mediante el uso del *widget* de diálogos de la librería jQuery UI. Permite abrir un cuadro de diálogo móvil y de tamaño variable para consultar el nombre y la pista asociada a la etapa. Para dar acceso al usuario a esta funcionalidad, se ha colocado en cada etapa un botón con el símbolo de característico de información.
- ❖ **Eliminar camino.** Permite eliminar un camino, borrando a su vez todas las etapas, geometrías, intentos de los usuarios hechos basados en el camino y respuestas a las preguntas de las etapas. Antes de proceder a eliminar se crea un cuadro de diálogo para que el usuario confirme su intención. Una vez confirmado, envía una petición AJAX a la función del servicio web encargada de eliminar caminos. Si el usuario dispone de permisos de “*Editar camino*” y el “*ticket*” del bloqueo de edición sigue siendo válido, el camino y todos los registros asociados son eliminados. Si no, la petición AJAX responderá con un error y no se eliminará ningún registro. Una vez que la petición AJAX responde satisfactoriamente, se borra de todos los contenedores de *features* la información relativa al camino eliminado. Para dar acceso al usuario a esta funcionalidad, se ha colocado en cada camino de la interfaz un botón con el símbolo característico de un cubo de basura.
- ❖ **Eliminar etapa.** Permite eliminar una etapa, borrando a su vez todas las geometrías, y respuestas a las preguntas de dicha etapa. Antes de proceder a eliminar se crea un cuadro de diálogo para que el usuario confirme su intención. Una vez confirmado, envía una petición AJAX a la función del servicio web encargada de eliminar etapas. Si el usuario dispone de permisos de “*Editar etapa*” y el “*ticket*” del bloqueo de edición sigue siendo válido, la etapa y todos los registros asociados son eliminados. Si no, la petición AJAX responderá con un error y no se eliminará ningún registro. Una vez que la petición AJAX responde satisfactoriamente, se borra de todos los contenedores de *features* la información relativa a la etapa eliminada. Para dar acceso al usuario a esta funcionalidad, se ha colocado en cada etapa de la interfaz un botón con el símbolo característico de un cubo de basura. Esta funcionalidad solo está disponible si el camino no está bloqueado, es decir, si ningún usuario/grupo ha realizado un intento sobre ese camino.
- ❖ **Crear camino.** Esta funcionalidad redirige al formulario de edición de caminos para configurar sus parámetros básicos. Si el usuario dispone de permisos de “*Añadir camino*”, se le mostrará el formulario. Si no, le aparecerá un mensaje de error y un botón de redirección a la página principal del módulo. Antes de abandonar la interfaz de edición intentará guardar automáticamente los cambios



producidos. Para dar acceso al usuario a esta funcionalidad, se ha colocado en el panel de herramientas de la interfaz un botón con el nombre “+ Camino”.

- ❖ **Crear etapa.** Esta funcionalidad redirige al formulario de edición de etapas para configurar sus parámetros básicos. Si el usuario dispone de permisos de “Añadir etapa”, se le mostrará el formulario. Si no, le aparecerá un mensaje de error y un botón de redirección a la página principal del módulo. Antes de abandonar la interfaz de edición intentará guardar automáticamente los cambios producidos. Para dar acceso al usuario a esta funcionalidad, se ha colocado en el panel de herramientas de la interfaz un botón con el nombre “+ Etapa”. Solo está disponible si el camino no está bloqueado, es decir, si ningún usuario/grupo ha realizado un intento sobre ese camino.
- ❖ **Editar camino.** Esta funcionalidad redirige al formulario de edición de caminos para modificar sus parámetros básicos. Si el usuario dispone de permisos de “Editar camino”, se le mostrará el formulario. Si no, le aparecerá un mensaje de error y un botón de redirección a la página principal del módulo. Antes de abandonar la interfaz de edición intentará guardar automáticamente los cambios producidos. Para dar acceso al usuario a esta funcionalidad, se ha colocado en cada camino de la interfaz un botón con el símbolo característico de un lápiz.
- ❖ **Editar etapa.** Esta funcionalidad redirige al formulario de edición de etapas para modificar sus parámetros básicos. Si el usuario dispone de permisos de “Editar etapa”, se le mostrará el formulario. Si no, le aparecerá un mensaje de error y un botón de redirección a la página principal del módulo. Antes de abandonar la interfaz de edición intentará guardar automáticamente los cambios producidos. Para dar acceso al usuario a esta funcionalidad, se ha colocado en cada etapa de la interfaz un botón con el símbolo característico de un lápiz.

Además de todas las funcionalidades, si no se han guardado los cambios realizados en las etapas y se pretende salir de la página de edición, se genera un cuadro de confirmación para el usuario indicando que las modificaciones en las etapas no se guardarán si continúa.

### **Formulario de edición de caminos**

El formulario de edición de caminos permite configurar los parámetros básicos de cada camino. Este se muestra cuando el profesor crea un nuevo camino disponiendo del permiso de “Añadir camino”, o bien al editar uno existente disponiendo del permiso de “Editar camino”.

Dentro del formulario de edición del camino se define una única pestaña llamada “General”, que debe contener los campos de la configuración del camino. Estos son:

- ❖ **Nombre del camino:** Un campo de tipo texto para introducir el nombre que se mostrará en la interfaz de edición y en la tabla de progreso general de los usuarios. Utilizando las funciones de formularios de Moodle se define el campo como obligatorio, y se añade una norma para que el campo no supere cierto número de caracteres.

- ❖ **Grupo asignado al camino / Agrupación asignada al camino:** Un campo de tipo selector que permite al profesor elegir qué grupo (si la opción de jugar en grupo está desactivada en la configuración general) o qué agrupación (si la opción está activada) del curso se asigna para realizar el camino al empezar el juego. En el caso de que la actividad esté configurada para jugar individualmente, no se seleccione ningún grupo y no existan más caminos en la actividad, permite participar a todos los usuarios matriculados con permiso para jugar. Este campo solo muestra aquellos grupos o agrupaciones que no han sido escogidos en ningún otro camino de la instancia. De esta manera se evita el error de seleccionar la misma opción en varios caminos.

Al final del formulario, se generan los botones de guardar cambios, que redirige a la pantalla de edición marcando como parámetro URL el camino en el que se encuentra, y cancelar, que desprecia cualquier dato introducido y redirige a la pantalla de edición sin ningún parámetro de camino.

### **Formulario de edición de etapas**

El formulario de edición de etapas permite configurar los parámetros básicos de cada etapa. Este se muestra cuando el profesor crea una nueva etapa disponiendo del permiso de “*Añadir etapa*”, o bien al editar una existente disponiendo del permiso de “*Editar etapa*”.

Para empezar, se define una pestaña “General”, que debe contener los campos más comunes de configuración de la etapa. Estos son:

- ❖ **Nombre de etapa:** Un campo de tipo texto para introducir el nombre que se mostrará en la interfaz de edición y de juego. Utilizando las funciones de formularios de Moodle se define el campo como obligatorio, y se añade una norma para que el campo no supere cierto número de caracteres.
- ❖ **Pista para localizar la siguiente etapa del camino:** Un campo de tipo *editor* que permite al profesor definir la pista proporcionada a los alumnos para descubrir la localización de la siguiente etapa. Se ha seleccionado explícitamente el campo *editor* para posibilitar la subida de contenido multimedia, como vídeos embebidos, audios, fotos, texto formateado o links hacia otras páginas, de modo que se pueda crear la pista más completa posible. Este campo se define como obligatorio.
- ❖ **Descubrir etapa sin desplazarse:** Un campo de tipo *checkbox* que permite al profesor definir si la etapa puede descubrirse sin tener que acudir presencialmente. Esta opción ha sido necesaria implementarla para conseguir que los jugadores puedan resolver etapas de difícil acceso, o con baja cobertura de datos o señal GPS, sin perder el método de juego de asistencia presencial en el resto de etapas del camino. Para ello, se sigue utilizando la marca sobre el mapa que el usuario quiere enviar como localización para descubrir la etapa. Una vez completada, el método de juego cambia al definido en las opciones generales de la actividad.

Uno de los requisitos globales del módulo definía que una etapa debería permitir asignar una actividad a completar y/o una pregunta simple a resolver antes de que se mostrara al usuario la pista asociada a la etapa. Para dar cabida a ello se define una pestaña de “Restricciones para descubrir la pista” que contiene los campos:

- ❖ **Completar antes la actividad seleccionada:** Un campo de tipo selector que permite al profesor elegir la actividad del curso a completar antes de que se muestre al alumno la pista actual. Para disponer de esta funcionalidad, se hace uso de la API de Moodle de finalización de actividad. Sin embargo, para que este campo recoja entre sus opciones una actividad, es necesario habilitar la opción de finalización de actividad dentro del panel de administración de Moodle, luego habilitarlo en el curso, después entrar en la pestaña “Finalización de la actividad” que se encuentra en el formulario de configuración de la actividad deseada, y por último seleccionar las condiciones a cumplir para ser categorizada como completada. Estas condiciones pueden variar de acuerdo al módulo, pero las generales permiten marcar que sea el alumno el que seleccione cuando está completa la actividad, o que el alumno haya sido calificado con una nota.
- ❖ **Añadir pregunta:** Un campo de tipo selector que permite al profesor elegir si incluye o no una pregunta simple antes de mostrar al alumno la pista actual. Como los campos de pregunta y respuesta no pueden aparecer si la elección es “No”, se ha creado un mecanismo que recarga la página al cambiar la opción seleccionada, sin perder los datos escritos en ella.
- ❖ **Pregunta:** Un campo de tipo editor que permite al profesor definir la pregunta proporcionada a los alumnos para descubrir la pista de la etapa. Este campo se define como obligatorio. Solo es visible si en el campo “Añadir pregunta” la respuesta es sí.
- ❖ **Elecciones:** Un campo de tipo editor por cada elección que permite al profesor definir cada posible respuesta a la pregunta anterior. Se implementa además un botón para añadir dos respuestas más al formulario cada vez que se pulsa, aunque solo se tienen en cuenta aquellas cuyos campos no estén vacíos. Para conseguirlo, se ha usado una de las funciones de la API de formularios para repetir un mismo bloque de elementos. Como requisito se establece que debe existir únicamente una respuesta marcada como correcta y mínimo 2 respuestas posibles no vacías, si no aparecerá un mensaje de error indicando el problema. Las elecciones solo son visibles si en el campo “Añadir pregunta” la respuesta es sí.

Al final del formulario, se generan los botones de guardar cambios, que redirige a la pantalla de edición marcando como parámetro URL el camino en el que se encuentra la etapa, y cancelar, que desprecia cualquier dato introducido y redirige a la pantalla de edición sin ningún parámetro de camino.

### **Diagrama de flujo**

Teniendo en cuenta todas las funcionalidades definidas y todos los aspectos del entorno de edición, se presenta a continuación de forma simplificada el diagrama de flujo correspondiente a la funcionalidad de edición de la caza del tesoro.

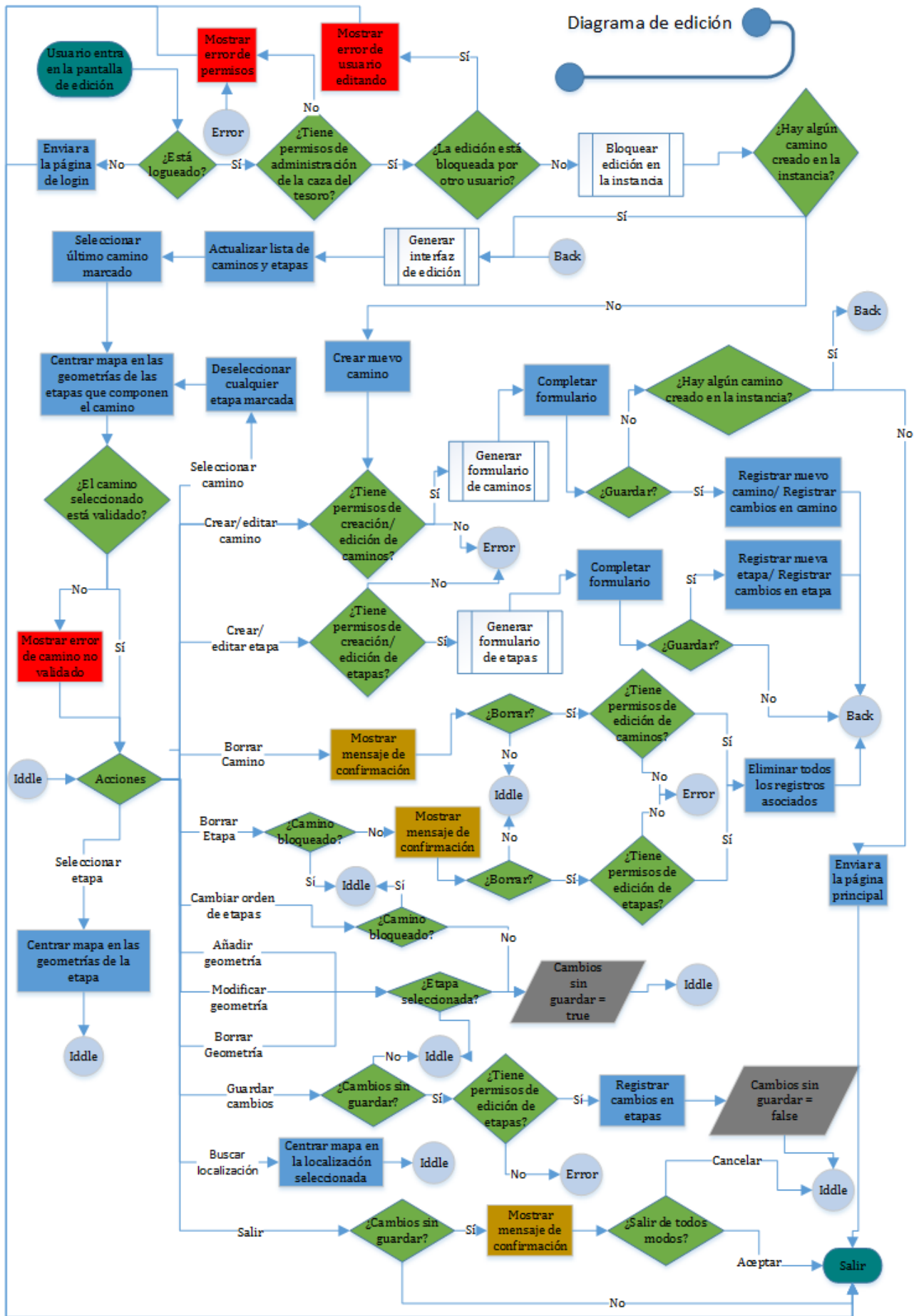


Figura 4.9 Diagrama de flujo de la funcionalidad de edición

## 4.2.2. Jugar

Esta funcionalidad está orientada al uso por parte del alumno, aunque es posible acceder a ella siempre que se disponga del permiso de jugar en el contexto de la actividad. Su finalidad es proporcionar una interfaz al alumno que le permita completar su camino asignado.

### Decisiones de diseño

Antes de comenzar a desarrollar el código para la implementación de la interfaz y funcionalidades del juego, se deben tener en cuenta diversos aspectos:

En primer lugar, al igual que ocurría en la interfaz de edición, es necesario garantizar en la interfaz de juego una interactividad fluida. Para ello se seguirá empleando el uso de la tecnología AJAX como medio de comunicación entre el navegador cliente y el servidor, de modo que no se tenga que recargar la página con cada nueva interacción del usuario con la interfaz de juego.

También es necesaria la gestión de datos espaciales, por lo que se seguirá empleando la misma solución aportada en la interfaz de edición: el uso de la librería Openlayers 3.

En segundo lugar, según lo definido en los requisitos globales del módulo, la interfaz de juego debe estar enfocada para dispositivos táctiles, es decir, dispositivos que suelen disponer de una pantalla relativamente pequeña. En estos casos, la simplificación es una materia clave para crear una experiencia instintiva y fácil de usar. Existen numerosos *frameworks* en el mercado que facilitan la creación de aplicaciones web para dispositivos táctiles. Por facilidad de uso y total compatibilidad con las tecnologías empleadas hasta el momento, se ha elegido jQuery Mobile [24]. Este *framework* está construido sobre la librería jQuery, y necesita de su enlazado en el documento web para funcionar. La forma de crear contenido sigue la estructura HTML, pero añadiendo a los contenedores (divs) los atributos *data-role*, que definen la funcionalidad y aspecto del elemento. Un div con un *data-role* de tipo *page* se considera un elemento de pantalla completa. El *framework* permite crear varios elementos de pantalla completa sobre un único documento HTML, de forma que descargando ese único documento se pueda acceder a todas las páginas que conforman una aplicación sin tener que descargar nuevo contenido. Así se garantiza que la interfaz tendrá un funcionamiento fluido e interactivo para el usuario.

Moodle, desde la versión 2.9, recomienda utilizar el lenguaje *mustache* [27] para crear bloques de contenido HTML personalizados. Este formato utiliza el mismo marcado que HTML, pero con etiquetas adicionales que se emplean para formatear la visualización de los datos. Con estas etiquetas es posible, entre otras cosas, realizar sustitución de variables por bloques de texto o bloques HTML, y la utilización de variables condicionales para que se imprima una cierta sección del documento o no. Este lenguaje se empleará para la construcción del marcado HTML necesario para la creación de nuevo contenido gráfico del *framework* jQuery Mobile.

En tercer lugar, la posibilidad del juego en modo grupos plantea un problema de sincronización. ¿Cómo actualizar la interfaz de juego de cada miembro del grupo de acuerdo a los intentos que realizan el resto de componentes del mismo? La solución propuesta incluye realizar peticiones AJAX periódicas a la función del servicio web del módulo que gestiona el progreso de los usuarios. En la petición, se envía la marca de tiempo del último intento registrado en la interfaz del cliente. Así, si se han producido cambios desde esa marca de tiempo, se actualizará la interfaz de juego y se notificará al usuario los intentos realizados por sus compañeros. Se deja en manos del administrador de Moodle el tiempo entre las peticiones AJAX, ya que existe un compromiso entre la rapidez de sincronización y la carga producida en el servidor.

En cuarto lugar, debido a que el módulo soporta los métodos de calificación de “puntuación por tiempo” y “puntuación por posición” existe una dificultad añadida a la hora de calcular las calificaciones, y es que la nota a partir del segundo puesto depende del número de usuarios/grupos que hayan finalizado la actividad. Por lo tanto, la solución pasa por recalcular las notas de todos los jugadores de la actividad cada vez que se produce una nueva finalización.

En quinto lugar, dado que el módulo soporta la realización de una actividad como requisito para superar una etapa, se hace necesario un método para comprobar si una actividad ha sido realizada por un usuario/grupo o no mientras se está dentro de la pantalla de juego. Gracias a la API de finalización de actividad que proporciona Moodle y a la solución propuesta anteriormente para la sincronización, se puede verificar si un usuario concreto ha concluido la actividad de acuerdo a la configuración marcada por el profesor. Además, cuando la realización de la actividad esté configurada para realizar por grupos, se ha decidido verificar si al menos uno de los componentes la ha finalizado para continuar, para mantener la misma filosofía que con las preguntas propuestas.

Por último, para la creación de la interfaz gráfica se debe tener en cuenta las partes que la componen. Para empezar, es necesario crear una pantalla principal de juego, dónde se muestre:

- ❖ Un mapa a pantalla completa con los intentos de localización del usuario, marcando los intentos satisfactorios con un icono diferente de los que representan los fallidos.
- ❖ Un panel de información general, dónde se muestre el nombre de usuario, la opción de salir de la interfaz de juego, un plegable con la descripción de la actividad y otro con la última pista descubierta.
- ❖ Un panel de búsquedas, que permita al usuario encontrar la localización requerida y centrarla en el mapa.
- ❖ Una barra de herramientas, que permita centrar el mapa en la localización del usuario, enviar la ubicación a comprobar y el acceso a otras pantallas secundarias.

También son necesarias una serie de pantallas secundarias: Una pantalla que permita cambiar el mapa base de la pantalla principal al usuario, otra que permita ver el historial de intentos del juego, y una última que permita mostrar una pregunta y sus respuestas.

## Funcionalidades incorporadas en la interfaz

Una vez definidas las tecnologías a emplear, la comunicación con el servidor, y las partes de las que consta la interfaz gráfica, es necesario concretar las funcionalidades que otorga el juego. A continuación, se detalla cada funcionalidad junto con los pasos dados para implementarla:

- ❖ **Mostrar panel de información.** Esta funcionalidad se ha desarrollado mediante el uso de *panel widget* y *collapsibleset widget* de la librería jQuery Mobile. Permite mostrar un panel en la parte izquierda de la pantalla con el nombre del usuario actual seguido de un botón para salir de la interfaz de juego, un plegable con el nombre y la descripción de la actividad y otro con el nombre actual de la etapa y la pista para llegar a la siguiente. Los plegables han sido diseñados para que solo se pueda abrir uno a la vez, cerrando el otro automáticamente con el fin de ahorrar espacio en la pantalla. En caso de que la pista de la etapa esté bloqueada, se mostrará en su lugar los pasos necesarios para desbloquearla. Puede estar bloqueada porque se necesite responder antes correctamente a una pregunta, en cuyo caso se creará un botón en la descripción que redirecciona a la pantalla de preguntas, también porque sea necesario completar antes una actividad del curso, en cuyo caso se indicará su nombre como enlace hacia su página de inicio, o ambas a la vez. Para dar acceso al usuario a esta funcionalidad, se ha creado en la interfaz un botón a la izquierda de la cabecera con el icono de un panel de opciones.
- ❖ **Mostrar panel de búsquedas.** Esta funcionalidad se ha desarrollado mediante el uso de las animaciones de la librería Openlayers 3, la utilización de *panel widget*, *filterable widget* y *listview widget* de jQuery Mobile y de la librería geocoder. El panel de búsquedas permite insertar una dirección o coordenadas, mostrando los resultados más aproximados. Su implementación sigue la misma línea que el buscador desarrollado en la interfaz de edición. Al seleccionar uno de los resultados se cierra el panel y centra el mapa en el lugar escogido. Para dar acceso al usuario a esta funcionalidad, se ha creado en la interfaz un botón a la derecha de la cabecera con el icono de una lupa.
- ❖ **Seleccionar intento.** Esta funcionalidad se ha desarrollado mediante el uso de los estilos de la librería Openlayers 3 y la utilización de *popup widget* de jQuery Mobile. Permite seleccionar un intento de localización mostrado en el mapa. Si es un intento satisfactorio, pueden ocurrir tres cosas: La primera, que se trate del intento desde donde el usuario descubrió la etapa actual, en cuyo caso se le abriría el panel de información mostrando la descripción de la pista o los pasos para desbloquearla. La segunda, que se trate del intento de una etapa anterior superada, en cuyo caso se mostrará al usuario un popup con el nombre, la pista y la fecha en la que fue superada. La tercera, que se haya sobrepasado el tiempo permitido para realizar la actividad y una pista se quede bloqueada, en cuyo caso el intento solo mostraría el *popup* con la fecha en la que descubrió la localización. Si el intento es fallido, al usuario se le mostrará un *popup* con la fecha en la que fue realizado. Para dar acceso al usuario a esta funcionalidad, se han creado sobre el mapa iconos

en la ubicación de cada intento con el número de la etapa a la que representan y que al seleccionarlos amplían su tamaño y resaltan su color.

- ❖ ***Autolocalizar***. Esta funcionalidad se ha desarrollado mediante el uso de los estilos y la geolocalización de la librería Openlayers 3. Permite al usuario conocer su ubicación, dibujando un punto negro y un área de aproximación en sus coordenadas y centrando el mapa alrededor ellas. Dependiendo de las opciones de geolocalización del navegador del usuario puede solicitarle permisos para compartir ubicación. En el navegador Chrome, existe el problema de que solo permite utilizar la geolocalización si la página tiene orígenes seguros (https). Para dar acceso al usuario a esta funcionalidad, se ha creado un botón en la barra de herramientas con típico icono de localización.
- ❖ ***Mostrar historial de intentos***. Esta funcionalidad se ha desarrollado mediante la utilización de *listview widget* de jQuery Mobile. Permite mostrar la pantalla de historial de intentos del usuario. Cada intento está antecedido de un icono de acierto o de fallo y describe lo ocurrido y en qué fecha. En el caso de que el modo de juego sea en grupos, especifica qué usuario realizó cada intento. Para dar acceso al usuario a esta funcionalidad, se ha creado un botón en la barra de herramientas con el típico icono de información.
- ❖ ***Mostrar selección de vistas del mapa***. Esta funcionalidad se ha desarrollado mediante la utilización de *listview widget* de jQuery Mobile y las propiedades de visibilidad de capas de Openlayers 3. Permite mostrar la pantalla de selección de vistas del mapa. Existen dos vistas de mapa dónde elegir: la vista de callejero, que emplea la capa estándar del mapa proporcionado por Open Street Map. O la vista aérea, que utiliza la capa aérea del mapa proporcionado por Bing Maps. La vista de callejero es la asignada por defecto al entrar en la interfaz de juego. Cuando el usuario selecciona un mapa, se cierra la pantalla de selección y el mapa base cambia al valor seleccionado. Para dar acceso al usuario a esta funcionalidad, se ha creado un botón en la barra de herramientas con el icono de lista de elementos.
- ❖ ***Validar ubicación***. Esta funcionalidad se ha desarrollado mediante la utilización de *popup widget* de jQuery Mobile y la geolocalización de Openlayers 3. Permite al usuario enviar su ubicación actual, si el método de juego es con desplazamiento, o la ubicación en el mapa de una bandera virtual (siempre que exista una bandera, si no se le pedirá al usuario que primero marque el punto deseado en el mapa), si el método de juego es sin desplazarse, para su correspondiente validación en el servidor. Antes de enviarla, se genera un *popup* de confirmación por si el usuario rectifica, ya que cada localización errónea puede bajar un porcentaje de la nota final si el profesor lo estipula en la configuración. Una vez confirmado, envía una petición AJAX a la función del servicio web encargada de comprobar la localización enviada. Si el usuario dispone de permisos de “Jugar” y la etapa no ha sido previamente resuelta por un compañero de equipo (si se juega en modo grupo), la localización se cotejará con la geometría de la etapa a descubrir y se registrará el intento como acierto o fallo, actualizando la calificación del usuario y devolviéndole un mensaje de aviso junto con todos los intentos realizados hasta el momento. Si, por un lado, el usuario no dispone de permisos de “Jugar”, la



petición AJAX responderá con un error y obligará al usuario a salir de la interfaz de juego. Si, por otro lado, un compañero de equipo ya había resuelto la localización de la etapa, no se tendrá en cuenta el intento y se devolverá al usuario las actualizaciones. Una vez que la petición AJAX responde satisfactoriamente, se actualizará la capa de intentos sobre el mapa y, en caso de que se haya acertado la posición, se abrirá el panel de información mostrando la pista o los pasos para desbloquearla. Esta funcionalidad solo está disponible cuando la pista no está bloqueada, es decir, si el usuario no tiene que superar ninguna actividad o pregunta para poder continuar, o cuando el tiempo para realizar la actividad no ha finalizado. Para dar acceso al usuario a esta funcionalidad, se ha creado un botón en la barra de herramientas con el nombre de “*Validar ubicación*”.

- ❖ **Responder pregunta.** Esta funcionalidad se ha desarrollado mediante la utilización de *controlgroup widget* de jQuery Mobile. Permite mostrar la pregunta asociada a la etapa actual y el conjunto de respuestas propuestas por el profesor. Cuando el usuario selecciona una respuesta y le da al botón de enviar, envía una petición AJAX a la función del servicio web encargada de comprobar la respuesta enviada. Si el usuario dispone de permisos de “Jugar” y la pregunta no ha sido previamente resuelta por un compañero de equipo (si se juega en modo grupo), la respuesta se comprobará y se registrará el intento como acierto o fallo, actualizando la calificación del usuario y devolviéndole un mensaje de aviso. Si, por un lado, el usuario no dispone de permisos de “Jugar”, la petición AJAX responderá con un error y obligará al usuario a salir de la interfaz de juego. Si, por otro lado, un compañero de equipo ya había resuelto la pregunta, no se tendrá en cuenta el intento y se devolverá al usuario las actualizaciones. Una vez que la petición AJAX responde satisfactoriamente, si la pregunta ha sido resuelta, cierra la pantalla de preguntas y desbloquea la pista en caso de que no haya una actividad por completar. Si la respuesta ha sido fallida, permanece en la pantalla de preguntas hasta que se envía una respuesta correcta. Esta funcionalidad solo está disponible cuando la etapa tiene una pregunta y aún no está resuelta. Para dar acceso al usuario a esta funcionalidad, se ha creado un botón dentro de los pasos para desbloquear la pista con el nombre de “Pregunta”.

### **Diagrama de flujo**

Teniendo en cuenta todas las funcionalidades definidas y todos los aspectos del entorno de juego, se presenta a continuación de forma simplificada el diagrama de flujo correspondiente a la funcionalidad de jugar de la caza del tesoro.

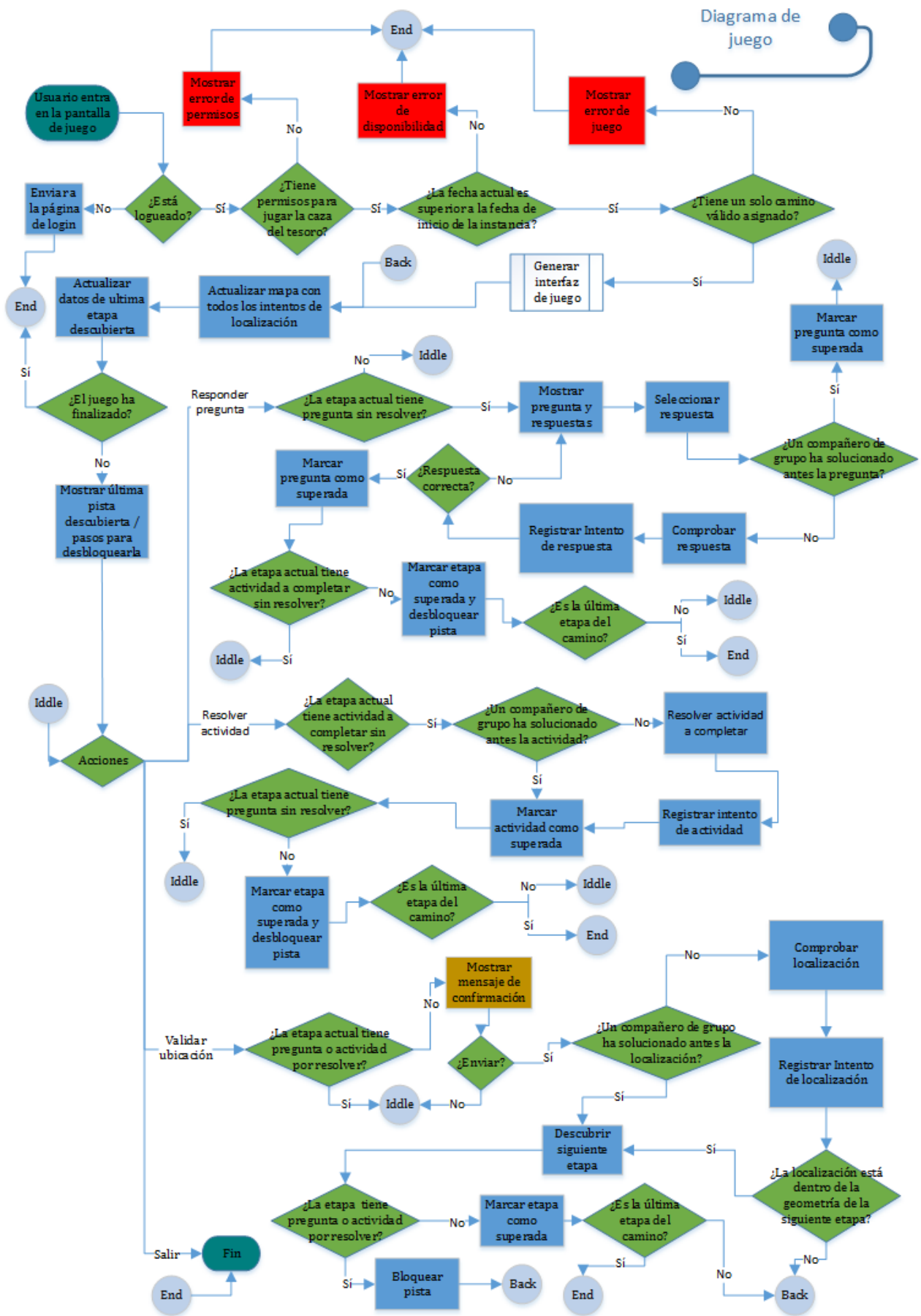


Figura 4.10 Diagrama de flujo de la funcionalidad de jugar

### 4.2.3. Otras funcionalidades

En este bloque se comentan todas las funcionalidades “secundarias” que proporcionan compatibilidad con Moodle y amplían las características de la aplicación:

Para empezar, se han incluido los eventos de inicio y finalización de la actividad de la caza del tesoro en el calendario de Moodle. Para ello, ha sido necesario una función que actualice los eventos que es llamada cada vez que se modifican las opciones generales de la actividad, y otra que los borre si la actividad es destruida.

También se ha incluido la posibilidad de reiniciar todos los datos de usuario cuando se reinicia un curso. Para implementarlo, ha sido necesario crear tanto la función que se encarga de crear el formulario de las opciones a borrar, como las funciones que al ser llamadas borran los intentos de los usuarios y las notas de todas las instancias del curso.

Asimismo, se ha incluido la posibilidad de realizar copias de seguridad de una actividad y restaurarla sin problemas en otra plataforma Moodle compatible. Para conseguirlo, ha sido necesario crear todos los ficheros que consultan el estado actual de la actividad, teniendo especial cuidado con aquellos atributos que necesitan un tratado especial, y también crear todos los ficheros encargados de restaurar la actividad a partir de la copia de seguridad.

Por último, era necesario un lugar dónde los alumnos pudieran ver de un vistazo su progreso particular como individuos o como grupo y observar su posición con respecto al resto, y también dónde los profesores pudiesen observar en tiempo real la situación de cada usuario y su histórico de intentos. De ello afloró el bloque de visualización de los resultados, centrado en la pantalla principal de la actividad para mayor accesibilidad. Este bloque, además de mostrar los resultados, se encarga de mostrar avisos tanto a alumnos como profesores:

- ❖ A los profesores con derecho de edición les avisa sobre fallos en las configuraciones de la caza, ya sea por un camino no validado, un camino sin ningún usuario/grupo asignado, o todos aquellos alumnos con permiso para jugar que bien porque están en más de un camino asignado, o bien porque no tienen asignado ningún camino no se les permite realizar la actividad.
- ❖ A los alumnos con permiso para jugar les avisa sobre fallos a la hora de jugar ya sea porque tienen asignado un camino no validado; tienen asignado más de un camino o no tienen asignado ninguno.

## 5. Manual de usuario

La presente sección pretende servir de guía para el uso del módulo. Para una mayor comprensión de las posibilidades que brinda para cada tipo de usuario, se ha dividido en función del rol que ingrese en la plataforma. Se ha distinguido principalmente entre el rol de administrador, encargado de instalar/desinstalar los módulos y configurar los valores por defecto, el profesor, encargado principalmente de la creación y diseño de la caza del tesoro, y los alumnos, encargados de completarla.

### 5.1. Manual para el administrador

#### 5.1.1. Instalación del módulo

Antes de instalar el módulo, debe asegurarse de que su versión de Moodle es igual o superior a la 2.9, si no al intentar continuar con la instalación fallará la revisión de dependencias. Para comprobar su versión actual de Moodle solo tiene que acceder a su sitio como administrador e ir a *Administración* → *Administración del sitio* → *Notificaciones*.

Moodle 3.1+ (Build: 20160603)  
Copyright © 1999 onwards, Martin Dougiamas  
and many other contributors.  
GNU Public License

*Figura 5.1 Versión de Moodle en la página de notificaciones*

Una vez cerciorado, podrá escoger una de las 3 maneras distintas de llevar a cabo la instalación del módulo de Moodle que se detallan a continuación [30].

##### 5.1.1.1. Instalación directamente desde el directorio de *plugins* de Moodle

- ❖ Asegúrese de que el directorio */mod* localizado dentro del directorio raíz de la instalación de Moodle tiene permisos de escritura.
- ❖ Acceda a su sitio Moodle como administrador y vaya a *Administración* → *Administración del sitio* → *Extensiones* → *Instalar módulos externos*.
- ❖ Elija el botón '*Instalar módulos externos desde el directorio de extensiones de Moodle*'.
- ❖ Busque el *plugin* que desea instalar, en este caso '*Treasurehunt*', elija el botón *Install* y luego presione *Continuar*.
- ❖ Revise que aparezca el mensaje de que pasó la validación y después presione el botón '*Continuar*'.
- ❖ Una vez instalado configure las opciones por defecto globales del módulo.

### 5.1.1.2. Instalación mediante un archivo ZIP subido al sitio

- ❖ Asegúrese de que el directorio /mod localizado dentro del directorio raíz de la instalación de Moodle tiene permisos de escritura.
- ❖ Acceda a su sitio Moodle como administrador y vaya a *Administración* → *Administración del sitio* → *Extensiones* → *Instalar módulos externos*.
- ❖ Suba el archivo ZIP que contiene el módulo ‘*Treasurehunt*’, bien desde el botón ‘*Seleccionar un archivo...*’ o bien arrastrándolo y soltando dentro del contenedor. Después presione el botón ‘*Instalar módulo externo desde archivo ZIP*’.
- ❖ Revise que aparezca el mensaje de que pasó la validación y después presione el botón ‘*Continuar*’.
- ❖ Usted será automáticamente redirigido hacia *Configuraciones* → *Administración del sitio* → *Notificaciones* y tendrá lugar el proceso de instalación
- ❖ Una vez instalado configure las opciones por defecto globales del módulo.

### 5.1.1.3. Instalación manual en el servidor

- ❖ Copie de manera manual el directorio del módulo ‘*treasurehunt*’ en el directorio /mod localizado dentro del directorio raíz de la instalación de Moodle.

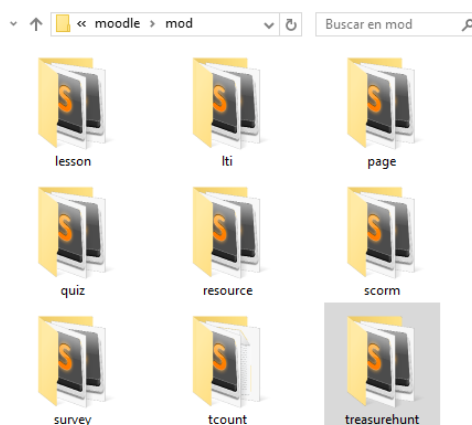


Figura 5.2 Instalación manual en el servidor

- ❖ Acceda a su sitio Moodle como administrador y vaya a *Administración* → *Administración del sitio* → *Notificaciones*.
- ❖ Una vez instalado configure las opciones por defecto globales del módulo.

## 5.1.2. Desinstalación del módulo

La desinstalación de cualquier módulo en Moodle tiene un procedimiento común que consiste en:

- ❖ Acceda al sitio como administrador, vaya a *Administración* → *Administración del sitio* → *Extensiones* → *Módulos de actividad* → *Gestionar actividades y*

presione el botón ‘*Desinstalar*’ en la fila de la tabla correspondiente al módulo que quiere desinstalar, en este caso ‘*Caza del tesoro*’.













 Archivo	0	2016052300		<a href="#">Configuración</a>	<a href="#">Desinstalar</a>
 Paquete SCORM	0	2016052300		<a href="#">Configuración</a>	<a href="#">Desinstalar</a>
 Encuestas predefinidas	0	2016052300			<a href="#">Desinstalar</a>
 Caza del tesoro	0	2016061209		<a href="#">Configuración</a>	<a href="#">Desinstalar</a>
 URL	0	2016052300		<a href="#">Configuración</a>	<a href="#">Desinstalar</a>
 Wiki	0	2016052300			<a href="#">Desinstalar</a>

Figura 5.3 Desinstalación del módulo

- ❖ Confirme la desinstalación.
- ❖ Una vez desinstalado confirme la eliminación del directorio del módulo para que no sugiera volver a instalarlo.

## 5.2. Manual para el profesor

### 5.2.1. Creación de una actividad

La creación de una actividad en Moodle tiene un procedimiento común:

- ❖ Acceda a la plataforma como profesor, seleccione el curso dónde desea crear la nueva actividad y habilite, si aún no lo está, la edición pulsando el botón “*Activar edición*”.
- ❖ Seleccione dentro de la semana dónde quiera crear la instancia el botón “*+Añadir una actividad o un recurso*”.
- ❖ Escoja la actividad “*Caza del tesoro*” y presione el botón “*Agregar*”.

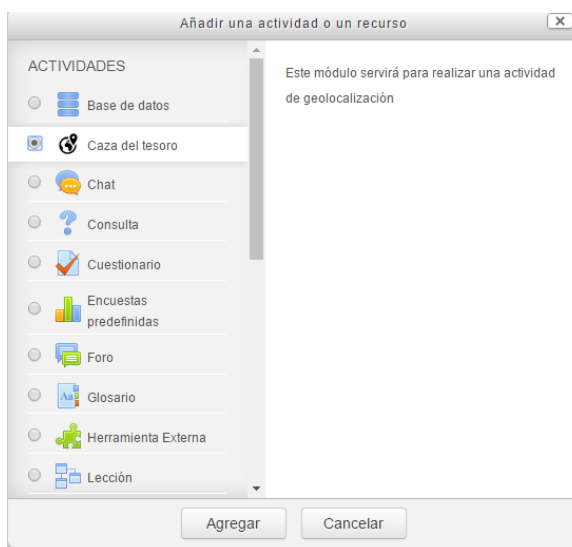


Figura 5.4 Creación de la actividad

- ❖ Aparecerá el formulario de configuración de la actividad que deberá rellenar y cuyo contenido se explica en el siguiente punto.

## 5.2.2. Pantalla de configuración de la actividad

La pantalla de configuración de la actividad permite establecer los parámetros básicos del juego. Es accesible desde distintos puntos: cuando se crea una nueva instancia de la caza del tesoro, cuando el usuario ha ingresado como profesor en el curso y estando habilitada la edición pulsa “*Editar*” → “*Editar ajustes*” al lado de una instancia de la caza del tesoro, o bien cuando el usuario ha ingresado como profesor en una instancia de la caza del tesoro y selecciona *Administración* → *Administración de la caza del tesoro* → *Editar ajustes*.

Dentro del formulario de configuración de la actividad se distinguen distintas pestañas con parámetros a rellenar:

### ❖ Pestaña **General**:

- ***Nombre de la caza del tesoro*** (Obligatorio).
- ***Descripción***.
- ***Mostrar la descripción en la página del curso***: Si se activa, la descripción anterior se mostrará en la página del curso justo debajo del enlace a la actividad o recurso.
- ***Jugar sin desplazarse***: Si esta opción se habilita los alumnos podrán jugar sin desplazarse a los lugares. Para ello cada vez que el alumno realiza un click simple sobre el mapa se crea una marca, borrando la anterior si existiese, indicando el último punto deseado.

### ❖ Pestaña **Disponibilidad**:

- ***Permitir intentos desde***: Si está habilitado, los estudiantes no podrán jugar antes de esta fecha. Si está deshabilitado, los estudiantes podrán comenzar a jugar de inmediato.
- ***Fecha límite***: Si se activa la opción, no se aceptarán intentos después de esta fecha sin una ampliación. Si la opción “*Permitir intentos desde*” está habilitada la fecha límite debe ser mayor, si no aparecerá un mensaje de advertencia y no permitirá guardar los cambios.
- ***Mostrar siempre la descripción***: Si está deshabilitado, la Descripción de la Caza del tesoro superior solo será visible para los estudiantes en la fecha “*Permitir intentos desde*”.

### ❖ Pestaña **Grupos**:

- ***Juego en grupos***: Si está habilitado los estudiantes se dividirán en grupos en función de la configuración de grupos del curso. El juego del grupo será compartido entre los miembros del grupo y todos ellos verán los cambios producidos en el juego.

❖ Pestaña **Calificación:**

- **Calificación:** Seleccione el tipo de calificación que desea utilizar para esta actividad. Si elige "*escala*", a continuación podrá elegirla de una lista desplegable. Si prefiere "*puntuación*", podrá elegir la puntuación máxima para esta actividad en un rango de valores enteros entre 0 y 100.
  - **Categoría de calificación:** Este parámetro controla la categoría en la que las calificaciones de esta actividad están ubicadas en el libro de calificaciones
  - **Calificación para aprobar:** Si un ítem tiene una calificación que los usuarios deben igualar o superar para aprobar un ítem, puede fijarla en este campo.
  - **Método de calificación:** Se habilita si la calificación está definida como "*puntuación*". Permite seleccionar el método de calificación que desea utilizar para esta actividad. Si elige puntuación por etapas, cada jugador (o equipo) puntúa de forma proporcional al número de etapas resueltas, siendo el 100% de la calificación máxima cuando se ha completado el camino y 0 cuando no se ha resuelto ninguna etapa. Si elige puntuación por tiempo, el ganador de la caza es el que marca el mejor tiempo. La calificación se calcula interpolando el tiempo de finalización, siendo el 50% de la calificación máxima el peor tiempo de finalización y el 100% el mejor. Los jugadores que no terminaron la caza reciben una calificación por debajo del 50% calculado simplemente por el número de etapas resueltas. Si elige puntuación por posición, la puntuación se calcula interpolando la posición en el ranking, siendo el 100% de la calificación máxima para al primer jugador (o equipo) en finalizar y 50% para el último jugador. Los jugadores que no terminaron la caza reciben una calificación por debajo del 50% calculado simplemente por el número de etapas resueltas.
  - **Penalización por fallo en localización:** Se habilita si la calificación está definida como "*puntuación*". La penalización es expresada en % de la calificación. Por ejemplo, si la penalización es 5.4, un jugador con 3 fallos penalizará su nota en un 16.2%, es decir, recibirá el 83.8% de la calificación calculada con el resto de criterios.
  - **Penalización por fallo en respuesta:** Se habilita si la calificación está definida como "*puntuación*". La penalización es expresada en % de la calificación. Por ejemplo, si la penalización es 5.4, un jugador con 3 fallos penalizará su nota en un 16.2%, es decir, recibirá el 83.8% de la calificación calculada con el resto de criterios.
- ❖ Otras pestañas comunes al resto de módulos, como **Ajustes comunes del módulo** dónde se configura la visibilidad de la actividad. También existen otras pestañas que aparecen dependiendo de la configuración de Moodle, como **Restricciones de acceso** que permite añadir restricciones que deben cumplirse para entrar en la



actividad o **Finalización de actividad** que permite especificar cuándo se considera la actividad como finalizada.

### 5.2.3. Pantalla de inicio

La pantalla de inicio es aquella que se presenta según se accede a una instancia de la caza del tesoro. Para el profesor se compone de distintas partes, entre las que podría estar “*Historial de intentos del profesor*”, pero como para ello se debe disponer del permiso de jugar propio de los alumnos se asume que no dispone de dicho permiso y se explica únicamente en el manual de alumnos.

#### 5.2.3.1. Información general de la caza del tesoro

En ella se muestra el título, la descripción, la fecha de apertura y cierre de la actividad en caso de existir, el modo de juego indicando si es sin desplazarse o en movimiento y el método de calificación configurado si la nota máxima es mayor que cero. Además, si la fecha de apertura es superior a la fecha actual muestra un botón de “*Volver al curso*”.

#### 5.2.3.2. Progreso de los usuarios

El progreso de los usuarios indica el avance de los alumnos/grupos de alumnos a través de las distintas etapas de los caminos asignados de la actividad en función de los colores: El color verde indica que la etapa se ha superado sin fallos, el amarillo que la etapa se ha superado con fallos, el rojo que la etapa no se ha superado y se han cometido fallos y el gris que la etapa no se ha superado y no se han cometido fallos. De esta forma el profesor puede ver de un vistazo en qué etapa se encuentra cada alumno/grupo. Además, para el rol de profesor tiene varias funciones adicionales:

- ❖ Muestra avisos de error (aunque solo para profesores con permisos de gestión de la caza del tesoro):
  - Avisa si todavía no se han añadido caminos a la actividad. Esta es la situación típica si se acaba de crear la instancia, y se soluciona accediendo a la edición de la caza del tesoro y procediendo como se explica en la siguiente sección.

## Caza del tesoro

La aventura de tu vida

Esta caza del tesoro está abierta desde el jueves, 2 de junio de 2016, 23:02

Método de calificación: Puntuación por pistas ?

### Progreso de los usuarios ?

Todavía no se han añadido caminos

Editar caza del tesoro

Figura 5.5 Pantalla de inicio al entrar por primera vez como profesor.

- Avisa debajo del nombre del camino correspondiente si este no está validado, es decir, si no tiene al menos dos etapas y que todas ellas tengan sus correspondientes geometrías de posición. Se soluciona pulsando el botón “*Editar caza del tesoro*” y procediendo como se explica en la siguiente sección.

#### Camino 1

El camino no está validado



Figura 5.6 Aviso de camino no validado

- Avisa en el caso de modo de juego individual debajo del nombre del camino correspondiente si ningún usuario tiene asignado dicho camino. Se soluciona pulsando en el botón “*Editar caza del tesoro*”

#### Camino 1

Ningún usuario asignado a este camino



Figura 5.7 Aviso de camino sin usuarios asignados

- Avisa en el caso de modo de juego en grupo debajo del nombre del camino correspondiente si ningún grupo tiene asignado dicho camino.

#### Camino 1

Ningún grupo asignado a este camino



Figura 5.8 Aviso de camino sin grupos asignados

- Avisa si existen alumnos matriculados al curso y con permiso para jugar que no tengan asignado ningún camino.

#### Progreso de los usuarios

Los siguientes usuarios no pertenecen a ningún grupo/agrupación: User 1, User 2, por lo que no son capaces de jugar la actividad.



Figura 5.9 Aviso de usuarios sin ningún camino asignado

- Avisa en el caso de modo de juego individual si existen alumnos que pertenezcan a más de un grupo y por lo tanto tengan asignados más de un camino.

## Progreso de los usuarios


Los siguientes usuarios pertenecen a más de un grupo: User 2, por lo que no son capaces de jugar la actividad. 

Figura 5.10 Aviso de alumnos con más de un camino asignado

- Avisa en el caso de modo de juego en grupo si existen grupos que pertenezcan a más de una agrupación y por lo tanto tengan asignados más de un camino.

## Progreso de los usuarios


Los siguientes grupos pertenecen a más de una agrupación: Grupo 1, por lo que no son capaces de jugar la actividad. 

Figura 5.11 Aviso de grupos con más de un camino asignado

- ❖ Permite seleccionar un usuario/grupo asignado a un camino y ver concretamente su historial de intentos (aunque solo para profesores con permisos de visión del historial de intentos de los usuarios), comprobando en qué apartados dentro de cada etapa ha fallado, como puede verse en la Figura 5.12.

## Caza del tesoro

Modo de juego: Jugar sin desplazarse

Método de calificación: Puntuación por etapas 

### Historial de intentos de **Grupo 1**

Intento	Estado
1	Localización encontrada por User 2 de la etapa 1 en la fecha: miércoles, 6 de julio de 2016, 19:44
2	Actividad a finalizar completada con éxito por User 1 para la etapa 1 en la fecha: miércoles, 6 de julio de 2016, 19:50
3	Respuesta acertada por User 1 a la pregunta de la etapa 1 en la fecha: miércoles, 6 de julio de 2016, 19:50
4	Etapas superadas por User 1 en la fecha: miércoles, 6 de julio de 2016, 19:50
5	Localización fallida por User 2 de la etapa 2 en la fecha: miércoles, 6 de julio de 2016, 19:51
6	Etapas superadas por User 2 en la fecha: miércoles, 6 de julio de 2016, 19:51

## Progreso de los usuarios

### Camino 1

Grupo	Etapas			
<b>Grupo 1</b>	1	2	3	4
Grupo 2	1	2	3	4

Editar caza del tesoro

Figura 5.12 Selección del historial de intentos de un usuario/grupo.

## 5.2.4. Pantalla de edición de la caza del tesoro

La pantalla de edición de la caza del tesoro permite crear, editar y eliminar caminos y las etapas que los contienen a través de la interfaz de la Figura 5.13. Es accesible desde distintos puntos siempre que se tengan permisos de gestión de la caza del tesoro: bien desde el botón “*Editar caza del tesoro*” disponible dentro de la pantalla de inicio o bien desde el panel de *Administración* → *Administración de la caza del tesoro* → *Editar caza del tesoro*.

### 5.2.4.1. Funcionalidades de la interfaz de edición

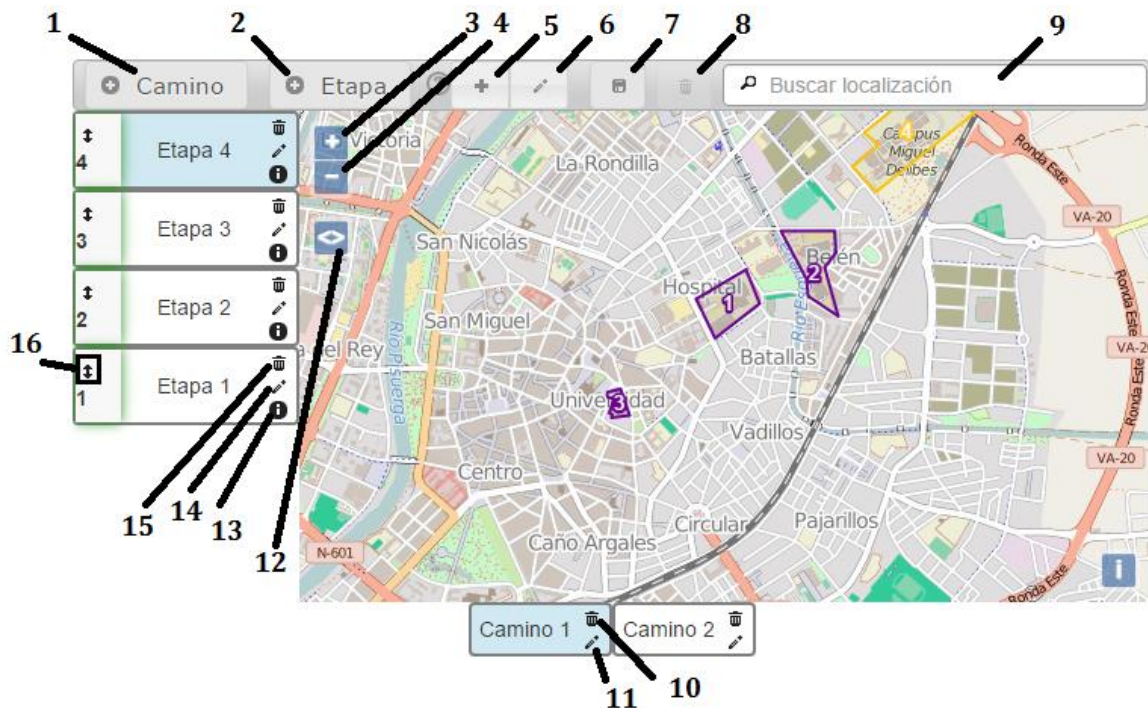


Figura 5.13 Interfaz de edición de la caza del tesoro

Dentro de la interfaz se distinguen distintas funcionalidades:

1. **Crear un nuevo camino.** Esta funcionalidad redirige a la pantalla de edición de caminos para configurar sus parámetros básicos, explicados en la sección 5.2.4.2. Requiere del permiso de “*Añadir camino*”.
2. **Crear una nueva etapa del camino seleccionado.** Esta funcionalidad redirige a la pantalla de edición de etapas para configurar sus parámetros básicos, explicados en la sección 5.2.4.3. Solo está disponible si el camino no está bloqueado, es decir, si ningún usuario/grupo ha realizado un intento sobre ese camino. Requiere del permiso de “*Añadir etapa*”.
3. **Aumentar el zoom del mapa.**
4. **Disminuir el zoom del mapa.**

5. **Añadir geometría a la etapa seleccionada<sup>4</sup>.** Para crear una nueva geometría debe hacer un *click* en el mapa por cada vértice de esta y cuando haya finalizado terminar con doble *click*. Puede interrumpir en cualquier momento la creación de la geometría pulsando la tecla “Esc”. Esta funcionalidad solo se habilita si existe una etapa seleccionada de la lista de etapas del camino elegido. Una vez añadida requiere la opción “*Guardar cambios*” para asegurar las modificaciones realizadas.
6. **Editar geometría de la etapa seleccionada.** Esta funcionalidad solo se habilita si existe una etapa seleccionada de la lista de etapas del camino elegido. Para editar la geometría es necesario seleccionarla clicando sobre ella en el mapa, lo que cambiará su color a azul, y luego arrastrar los vértices hacia la posición deseada. Una vez editada requiere la opción “*Guardar cambios*” para asegurar las modificaciones realizadas.
7. **Guardar cambios.** Guarda los cambios realizados en las posiciones y geometrías de las etapas. Requiere del permiso de “*Editar etapa*” para que los cambios se guarden con éxito.
8. **Eliminar la geometría seleccionada.** Al pulsar sobre esta funcionalidad aparecerá un cuadro de diálogo de confirmación indicando que no es posible recuperar la geometría una vez borrada. Esta funcionalidad solo se habilita si existe una geometría seleccionada, y para seleccionarla es necesario estar dentro de la opción “*Editar geometría de la etapa seleccionada*”. Una vez borrada requiere la opción “*Guardar cambios*” para asegurar las modificaciones realizadas.
9. **Buscador que otorga resultados de localización.** Al seleccionar uno de los resultados encuadra el mapa en la ubicación deseada.



Figura 5.14 Buscador de localización en la interfaz de edición.

<sup>4</sup> La geometría de la primera pista sirve como lugar de partida y será revelado a los alumnos al comenzar su camino indicándoles que solo pueden comenzar desde la geometría marcada en el mapa.

10. **Eliminar camino.** Al pulsar sobre esta funcionalidad aparecerá un cuadro de diálogo de confirmación indicando que no es posible recuperar el camino y todas las etapas que lo contienen una vez borrado. Requiere del permiso de “*Editar camino*”.
11. **Editar camino.** Esta funcionalidad redirige a la pantalla de edición de caminos explicada en la sección 5.2.4.2. Requiere del permiso de “*Editar camino*”.
12. **Cambiar relación de tamaño horizontal entre el contenedor de etapas y el mapa.** Esta funcionalidad solo está habilitada si existen etapas en el camino seleccionado.

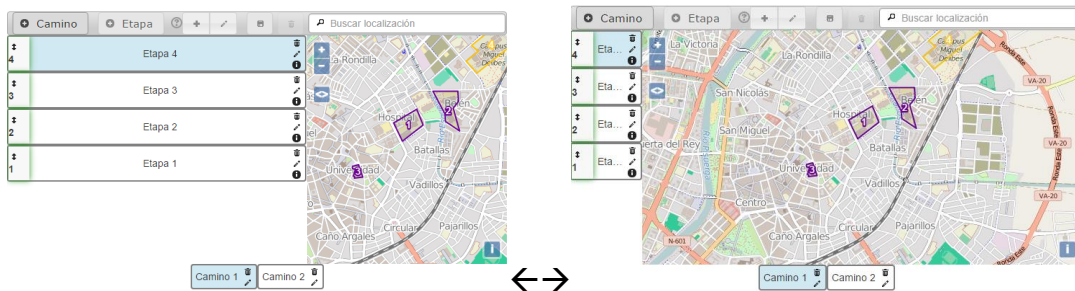


Figura 5.15 Cambio de relación de tamaño entre el contenedor de etapas y el mapa

13. **Mostrar el nombre y la pista de la etapa seleccionada.** Esta funcionalidad abrirá un cuadro de diálogo móvil y de tamaño variable para consultar los datos de la etapa.

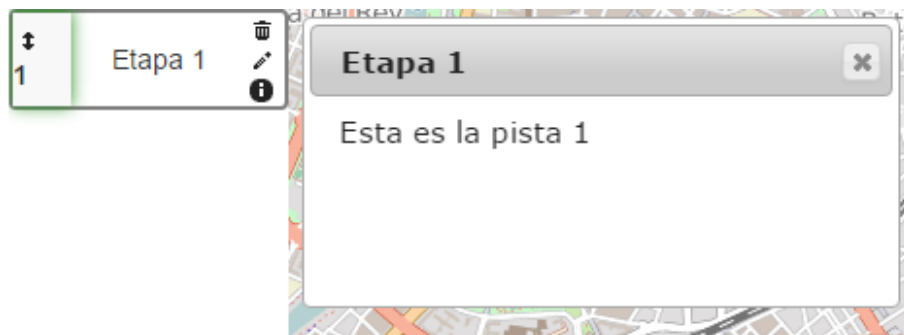


Figura 5.16 Cuadro de diálogo con información de la etapa.

14. **Editar etapa.** Esta funcionalidad redirige a la pantalla de edición de etapas explicada en la sección 5.2.4.35.2.4.2. Requiere del permiso de “*Editar etapa*”.
15. **Borrar etapa.** Al pulsar sobre esta opción aparecerá un cuadro de diálogo de confirmación indicando que no es posible recuperar la etapa una vez borrada. Esta funcionalidad solo está disponible si el camino no está bloqueado, es decir, si ningún usuario/grupo ha realizado un intento sobre ese camino, como se aprecia en la Figura 5.17. Requiere del permiso de “*Editar etapa*”.



Figura 5.17 Etapa de un camino bloqueado.

16. **Cambiar el orden de las etapas.** Para cambiar el orden de las etapas es necesario arrastrar cada una hacia la posición deseada. Requiere la opción “*Guardar cambios*” para asegurar las modificaciones realizadas. Esta funcionalidad solo está disponible si el camino no está bloqueado, es decir, si ningún usuario/grupo ha realizado un intento sobre ese camino, si no aparecerá un icono con un candado en lugar de las flechas como el de la Figura 5.17.

La interfaz pretende ser lo más intuitiva posible, intentando guiar al profesor en la creación de caminos válidos. Para ello cuenta con:

- ❖ Si se entra en la pantalla de edición de la caza del tesoro y no existe ningún camino redirige a la pantalla de edición de caminos para tener un primer camino con el que trabajar.
- ❖ Si no tiene al menos dos etapas se indica en un mensaje de error y se resalta el botón que permite crear una nueva etapa.



Figura 5.18 Interfaz de edición con un camino que tiene menos de 2 etapas.

- ❖ Si una etapa no tiene ninguna geometría asociada, se indica en un mensaje de error, se remarca en rojo dicha etapa y al seleccionarla se resalta el botón que permite crear una nueva geometría.



Figura 5.19 Interfaz de edición con una etapa sin geometría.

- ❖ Si no se han guardado los cambios realizados en las etapas y se pretende salir de la página de edición muestra un cuadro de confirmación indicando que las modificaciones en las etapas no se guardarán.

Además de estas funcionalidades, la pantalla de edición de la caza del tesoro cuenta con un **bloqueo de edición** que solo permite editar a un profesor a la vez para garantizar la fiabilidad e integridad de los cambios.

#### 5.2.4.2. Pantalla de edición de caminos

La pantalla de edición de caminos permite configurar los parámetros básicos de cada camino. Es accesible desde la interfaz de la pantalla de edición de la caza del tesoro, bien al crear un nuevo camino disponiendo del permiso de añadir camino o bien al editar uno existente disponiendo del permiso de editar camino.

Dentro del formulario de edición del camino se encuentra la pestaña **General** que contiene los siguientes parámetros a rellenar:

- ❖ **Nombre del camino** (Obligatorio).
- ❖ **Grupo asignado al camino / Agrupación asignada al camino**: Los usuarios/grupos de este/a grupo/agrupación son asignados a este camino cuando empieza el juego. Si la actividad está configurada para jugar individualmente se deberá asignar un grupo al camino. En el caso de que no se asigne ningún grupo y no existan más caminos en la actividad permite jugar a todos los usuarios matriculados con permiso para jugar. Si la actividad está configurada para jugar en grupos deberá asignarse una agrupación cuyos grupos representan los equipos en los que se dividirán los usuarios para jugar ese camino.



### 5.2.4.3. Pantalla de edición de etapas

La pantalla de edición de etapas permite configurar los parámetros básicos de cada etapa. Es accesible desde la interfaz de la pantalla de edición de la caza del tesoro, bien al crear una nueva etapa disponiendo del permiso de añadir etapa o bien al editar una existente disponiendo del permiso de editar etapa.

Dentro del formulario de edición de etapas se distinguen distintas pestañas con parámetros a rellenar:

❖ Pestaña **General**:

- **Nombre de la etapa** (Obligatorio): Se recomienda que identifique el lugar donde se encuentra esta etapa.
- **Pista para localizar la siguiente etapa del camino** (Obligatorio): Aquí se debe describir la pista para alcanzar la siguiente localización. En el caso de que esta sea la última etapa debe contener un mensaje de retroalimentación indicando que la caza del tesoro ha finalizado.
- **Descubrir etapa sin desplazarse**: Si esta opción se habilita, los alumnos podrán descubrir esta etapa sin tener que desplazarse. Para ello, se crea una marca en el último lugar clicado del mapa. Solamente funciona durante la realización de esta etapa, luego se seguirá jugando con la configuración por defecto de la actividad.

❖ Pestaña **Restricciones para descubrir la pista**: Se puede indicar qué condiciones tienen que superar los participantes antes de que se les muestre la pista para buscar la siguiente localización.

- **Completar antes la actividad seleccionada**: La actividad seleccionada deberá completarse antes de que se muestre la pista actual. Para que las actividades del curso se muestren en la lista debe estar habilitada la terminación de actividad en la configuración de Moodle, en el curso y en la propia actividad.
- **Añadir pregunta**: Permite añadir una pregunta sencilla antes de mostrar la descripción de la pista.
- **Pregunta**: Se habilita y visualiza si la opción “Añadir pregunta” tiene seleccionado “Sí”. (Obligatorio).
- **Elección N°**: Se habilita y visualiza si la opción “Añadir pregunta” tiene seleccionado “Sí”. Aquí se deben escribir las posibles respuestas que le aparecerán al usuario al intentar contestar la pregunta. Hay un botón para añadir dos respuestas más al formulario cada vez que se pulsa, pero solo se tendrá en cuenta aquellas cuyos campos no estén vacíos. Debe existir únicamente una respuesta marcada como correcta y mínimo 2 respuestas posibles no vacías, si no aparecerá un mensaje de error indicando el problema.

## 5.3. Manual para el alumno

### 5.3.1. Pantalla de inicio

La pantalla de inicio es aquella que se presenta según se accede a una instancia de la caza del tesoro. Para el alumno se compone de distintas partes:

#### 5.3.1.1. Información general de la caza del tesoro

En ella se muestra el título, la descripción, la fecha de apertura y cierre de la actividad en caso de existir, el modo de juego indicando si es sin desplazarse o en movimiento y el método de calificación configurado por el profesor si la nota máxima es mayor que cero. Además, si la fecha de apertura es superior a la fecha actual muestra un botón de “*Volver al curso*”.

#### 5.3.1.2. Historial de intentos del usuario/grupo

En el historial aparecen en verde aquellos intentos que se han resuelto con éxito y en rojo los fallidos. Si el modo de juego es en grupo informa del nombre del usuario que ha realizado el intento, independientemente de si es un fallo o un acierto. Además, debajo del historial de intentos se muestra el botón de “*Jugar*” en el caso de que la caza del tesoro no haya finalizado o no se haya alcanzado la fecha límite, y el de “*Revisión del juego*” en caso contrario. Ambos redirigen a la pantalla de juego, aunque en el primer caso se podrá seguir jugando y en el segundo simplemente observar las localizaciones enviadas sobre el mapa.

El historial de intentos no se muestra en los siguientes casos:

- ❖ Si la fecha de apertura de la actividad es superior a la fecha actual.
- ❖ Si el alumno o su grupo no tiene asignado un camino, lo que le devolverá un mensaje de error indicándolo.
- ❖ Si el alumno o su grupo tienen asignado más de un camino, lo que le devolverá un mensaje de error indicándolo.
- ❖ Si el alumno o su grupo tienen asignado un camino no validado, lo que le devolverá un mensaje de error indicándolo.
- ❖ Si el alumno no tiene permisos para jugar la caza del tesoro.

#### 5.3.1.3. Progreso de los usuarios

El progreso de los usuarios indica el avance de los alumnos/grupos de alumnos a través de las distintas etapas de los caminos asignados de la actividad en función de los colores: El color verde indica que la etapa se ha superado sin fallos, el amarillo que la etapa se ha superado con fallos, el rojo que la etapa no se ha superado y se han cometido fallos y el gris que la etapa no se ha superado y no se han cometido fallos. Es un componente diseñado para potenciar la competitividad, ya que dependiendo del método de calificación

seleccionado por el profesor el orden/tiempo de finalización es un factor muy importante en la nota.

### 5.3.2. Pantalla de juego

La pantalla de juego de la caza del tesoro permite realizar un camino asignado descubriendo todas sus etapas gracias a la interfaz de juego de la Figura 5.20. Es accesible desde el botón “Jugar” disponible dentro de la pantalla de inicio.

#### 5.3.2.1. Funcionalidades de la interfaz del juego



Figura 5.20 Interfaz de juego de la caza del tesoro

Dentro de la interfaz de juego se distinguen distintas funcionalidades:

1. **Abrir panel de información.** El panel de información se abre automáticamente al iniciar la pantalla de juego, y muestra el nombre del usuario actual seguido de un botón para salir a la pantalla de inicio, un plegable con la descripción de la actividad y otro con la descripción para llegar a la siguiente etapa junto con el número de etapa actual del total alcanzado. En caso de que la pista esté bloqueada aparecerá en la descripción los pasos a seguir para desbloquearla. El panel se cierra desde el botón superior izquierdo o bien pulsando fuera del mismo.

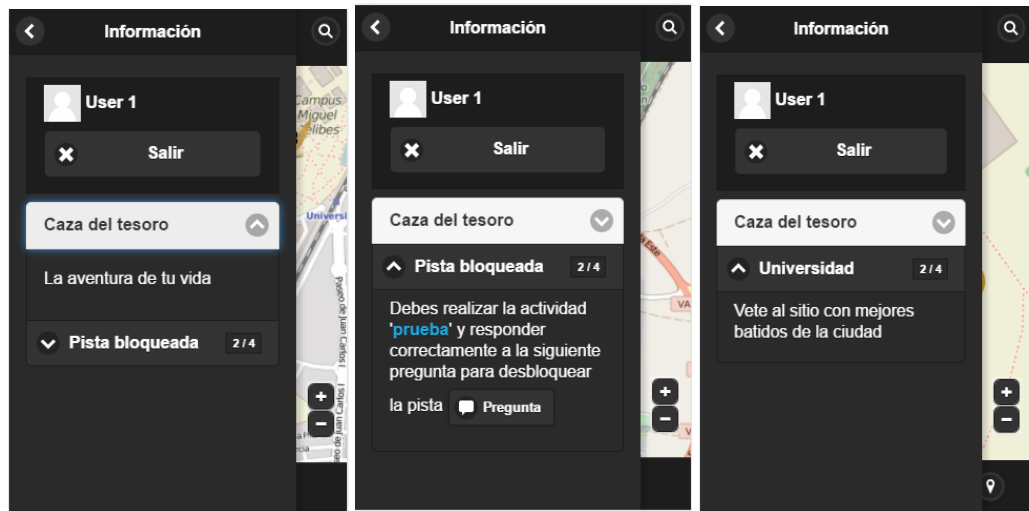


Figura 5.21 Panel de información con la descripción de la actividad, con una pista bloqueada y con otra sin bloquear.

2. **Abrir panel de búsquedas.** El panel de búsquedas muestra los resultados a una consulta de localización. Seleccionando uno de los resultados se cierra el panel y centra el mapa en el lugar escogido. Para cerrar el panel sin realizar una búsqueda pulse el botón superior derecho o bien pulse fuera del propio panel.

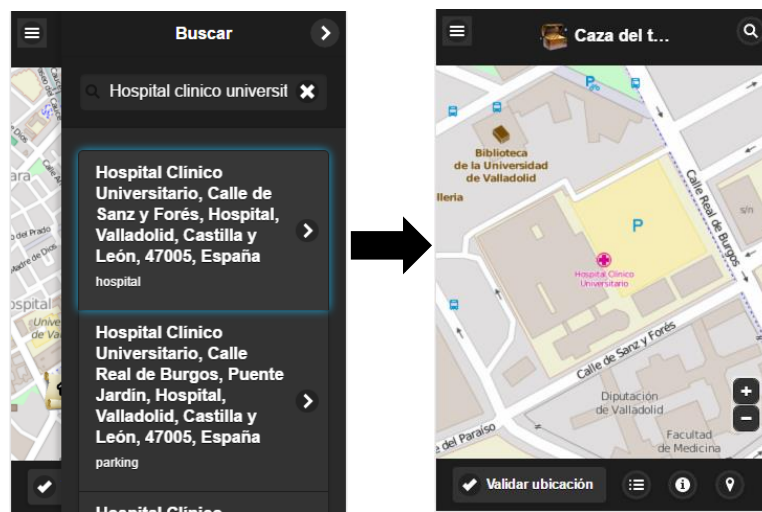


Figura 5.22 Panel de búsquedas de localización

3. **Abrir información de ubicación descubierta.** Al pulsar sobre el icono de una localización descubierta, muestra un *popup* con la información de la etapa y la fecha en la que fue descubierta, a no ser que se trate de la etapa actual y el camino aún no esté completado, en cuyo caso se le abrirá el panel de información con la descripción de la pista o los pasos para desbloquearla.

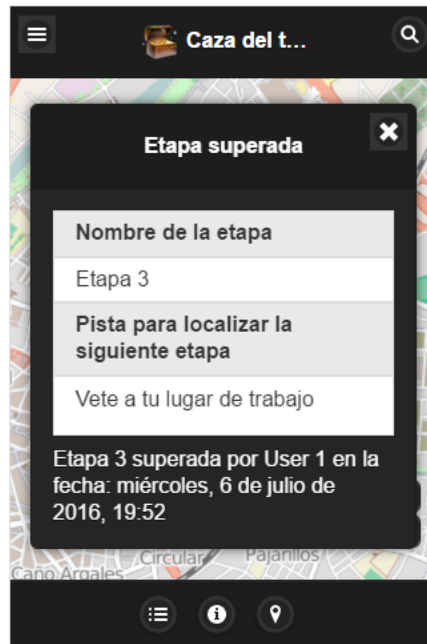


Figura 5.23 Cuadro de diálogo con información sobre una etapa superada.

4. **Abrir información de ubicación fallida.** Al pulsar sobre el icono de una localización fallida muestra un *popup* con la fecha en la que se falló la ubicación.



Figura 5.24 Cuadro de diálogo con la fecha sobre una localización fallida.

5. **Aumentar el zoom del mapa.**
6. **Disminuir el zoom del mapa.**
7. **Ubicar nuestra localización.** Al pulsarlo el mapa se centra en torno a la ubicación del usuario, dibujando un punto negro y un área de aproximación como el que se muestra en la Figura 5.25. Puede solicitar permisos para compartir ubicación.

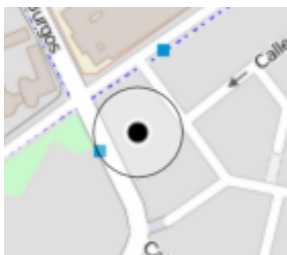


Figura 5.25 Icono de la última ubicación del usuario.

8. **Abrir pantalla de historial de intentos.** La pantalla de historial de intentos muestra el mismo historial que aparece en la pantalla principal, pero adaptado para no tener que salir del juego cada vez que se quiera consultarlo.

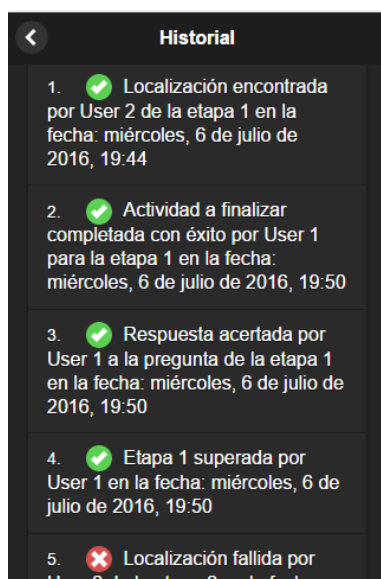


Figura 5.26 Pantalla de historial de intentos en el juego

9. **Abrir pantalla de selección de vistas del mapa.** La pantalla de selección de vistas del mapa permite seleccionar entre la vista en callejero que se asigna por defecto y la vista aérea. Al seleccionar uno se cierra para mostrar la pantalla de juego.

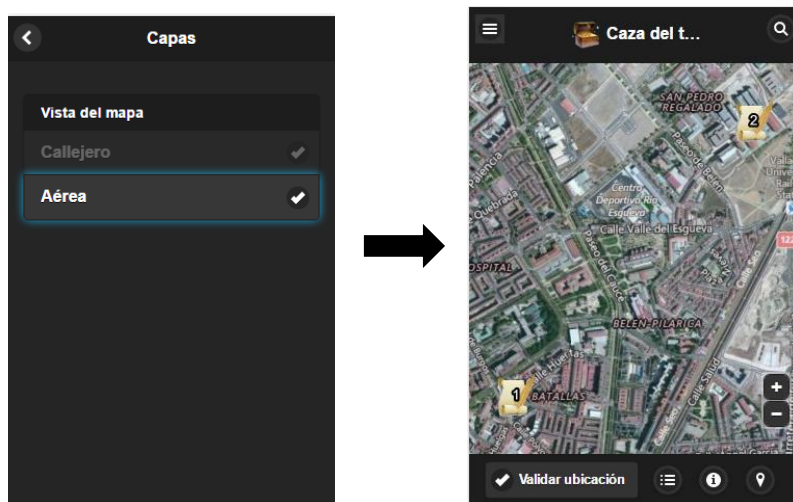


Figura 5.27 Pantalla de selección de vista del mapa en el juego.

10. **Validar ubicación.** Permite enviar la ubicación actual para su correspondiente validación en el servidor. Antes de enviarla aparece un cuadro de confirmación por si el usuario rectifica, ya que cada localización errónea puede bajar un porcentaje de la nota final si el profesor lo estipula en la configuración. Dependiendo del modo de juego la localización enviada será una u otra. En el caso de que el modo de juego sea sin desplazarse se enviará la ubicación de la bandera de localización virtual (siempre que exista una bandera, si no se le pedirá al usuario que primero marque un punto en el mapa), y en caso contrario la localización actual del usuario que mayor precisión dé empleando técnicas GPS/localización de IP o red móvil (puede solicitar permisos para compartir ubicación). Posteriormente se recibirá del servidor una respuesta indicando si se ha acertado o no, y marcará el intento en el mapa con un icono de acierto o fallo. En caso de acierto se abrirá el panel de información mostrando la descripción de la pista o los pasos a seguir para desbloquearla. Los fallos en la localización no se tendrán en cuenta si estando en modo grupo un miembro descubre la ubicación de la etapa actual justo antes de que se produzcan los fallos de sus compañeros. Esta funcionalidad solo está habilitada cuando la pista no está bloqueada, es decir, si no hay que superar ninguna actividad o pregunta para poder continuar, o cuando no se ha entrado en el modo revisión de juego explicado en 5.3.2.3.



Figura 5.28 Proceso de envío y validación de una ubicación en el juego

11. **Bandera de localización virtual.** Como se ha explicado, la bandera de localización virtual solo está habilitada cuando el modo de juego es sin desplazarse. En un primer momento no hay ninguna bandera en el mapa, debe marcarse el punto deseado y luego emplear la opción “Validar ubicación”. Cada vez que se marca un punto en el mapa la bandera se traslada a esa posición.
12. **Abrir pantalla de preguntas.** Muestra la pregunta asociada a la etapa actual y el conjunto de respuestas propuestas por el profesor. Se debe seleccionar una respuesta y darle al botón de “Enviar”. El servidor comprueba si la respuesta es la correcta y envía un mensaje indicándolo. Si es correcta cierra la pantalla de preguntas y marca la pregunta como acertada, desbloqueando la pista en el caso de que no haya una actividad por completar. Si es falsa permanece en la pantalla

de preguntas hasta que se envíe una respuesta correcta. En caso de darle al botón de “*Cancelar*” vuelve a la pantalla de juego. Los fallos en la respuesta no se tendrán en cuenta si estando en modo grupo un miembro acierta la respuesta correcta de la pregunta actual justo antes de que se produzcan los fallos de sus compañeros.

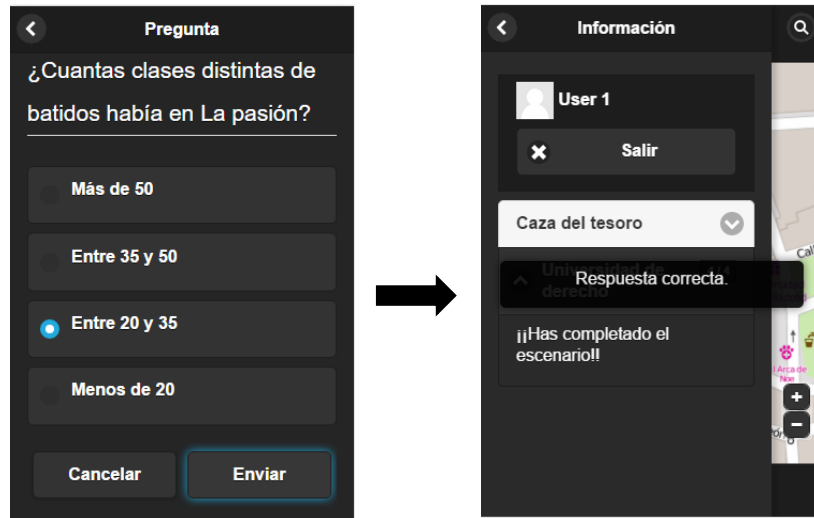


Figura 5.29 Proceso de envío y validación de la respuesta a una pregunta en el juego.

### 5.3.2.2. Primera etapa

La forma de comenzar un camino es especial, ya que pretende que el usuario empiece desde un lugar concreto. Para ello, se le indica en el panel de información que solo descubrirá la primera etapa si comienza desde dentro del área (la geometría) marcada y encuadrada en el mapa, ya que se considera el lugar de partida para emprender el camino. En el caso de que exista más de una geometría, el usuario puede empezar desde cualquiera de ellas.

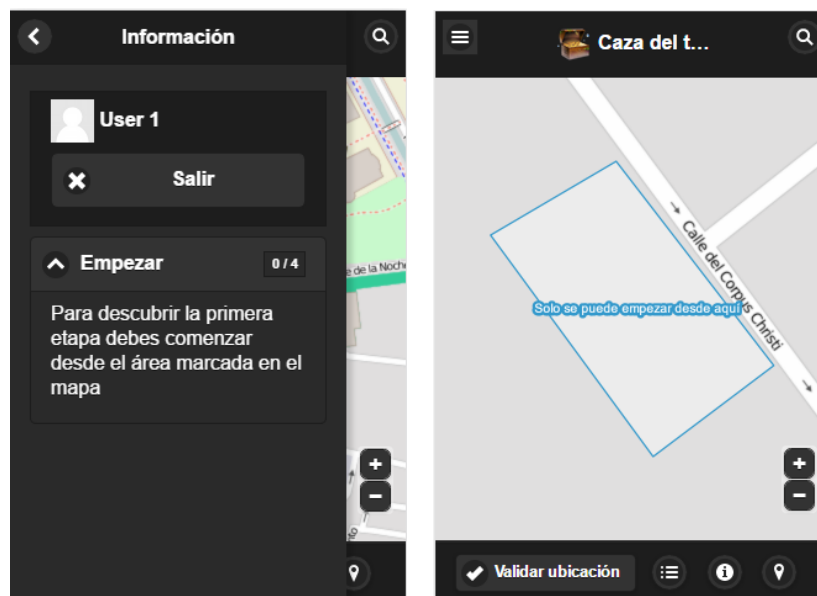


Figura 5.30 Proceso para descubrir la primera etapa.



### 5.3.2.3. Modo revisión del juego

El modo revisión de juego se alcanza en dos circunstancias: bien cuando el usuario ha completado el camino o bien cuando el profesor ha configurado un tiempo límite para completar la actividad y este se ha superado. En este modo no se permite realizar ningún tipo de intento, ni de localización ni respuesta de preguntas, y solo sirve para poder observar los intentos de ubicación enviados durante la realización de la actividad y abrir la información asociada a cada etapa.

### 5.3.2.4. Notificaciones

Durante el transcurso del juego pueden saltar notificaciones por distintos motivos:

- ❖ **Cambios producidos por los compañeros de grupo:** Este tipo de notificaciones ocurren cuando el modo de juego es por equipos y un miembro del grupo realiza un intento, ya sea de localización o de respuesta a una pregunta. A partir de ahí todos los demás usuarios del grupo que estén conectados reciben un aviso de actualización con el resultado del intento/s de su compañero.

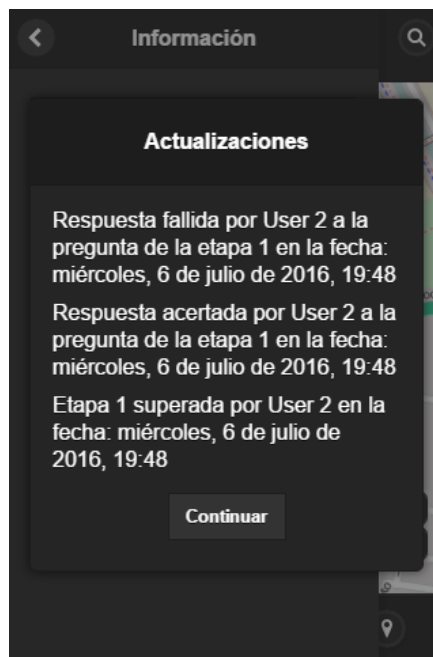


Figura 5.31 Notificación por cambios producidos por los compañeros de equipo.

- ❖ **Cambios en el modo de juego:** Estas notificaciones se producen cuando el profesor cambia el valor del modo de juego en grupos o el de jugar sin desplazarse. En el primer caso, se envía un aviso de actualización a los usuarios conectados en la pantalla de juego indicando si a partir de entonces jugarán individual o por equipos, cambiando a su vez los intentos descubiertos y fallidos en ese camino y modo de juego. En el segundo caso se envía un aviso de actualización a los usuarios indicando si jugarán con o sin desplazarse, y en caso de que el modo de juego anterior fuese sin desplazamiento se elimina la bandera de ubicación virtual.

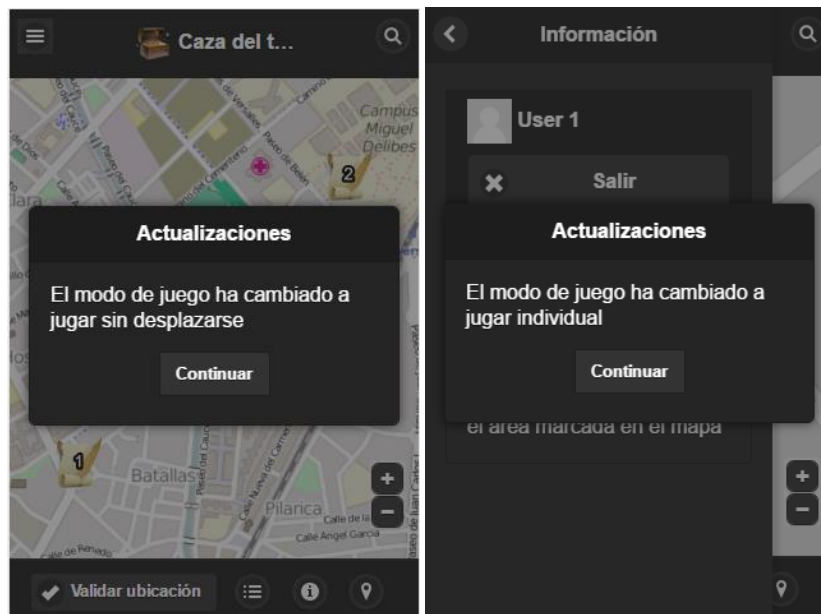


Figura 5.32 Notificaciones por cambios en el modo de juego.

- ❖ **Completar una actividad:** Estas notificaciones se producen cuando un usuario/grupo completa una actividad. Como el requisito de completar una actividad para desbloquear la pista se realiza fuera de la interfaz del juego, al registrar que se ha completado se envía un aviso al usuario/grupo conectados en ese momento indicando que se ha superado.

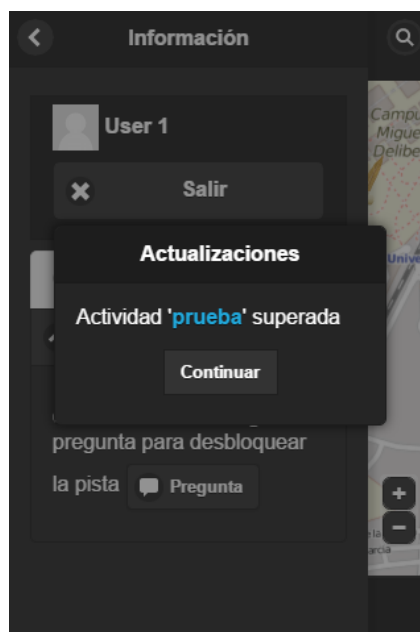


Figura 5.33 Notificación de actividad a completar superada.

- ❖ **Fuera de tiempo:** Estas notificaciones se producen cuando finaliza el tiempo configurado por el profesor para completar la actividad, en cuyo caso se envía al usuario/grupo un aviso de que se ha superado el tiempo y que entra en modo revisión del juego.

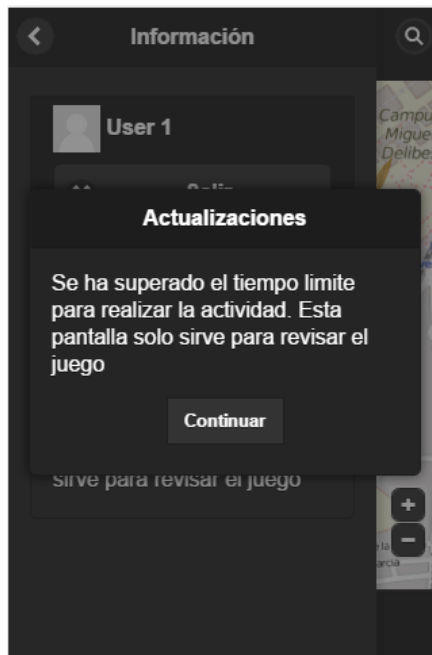


Figura 5.34 Notificación de tiempo superado para realizar la actividad.

- ❖ **Errores:** Estas notificaciones se producen cuando ocurre un error durante el juego, como la pérdida de respuesta con el servidor o en los extraños casos de que un profesor cambie los grupos/agrupaciones o borre el camino actual del usuario, en cuyo caso el alumno dejaría de tener un camino asignado. Obligan al usuario a redirigirse a la pantalla de inicio.

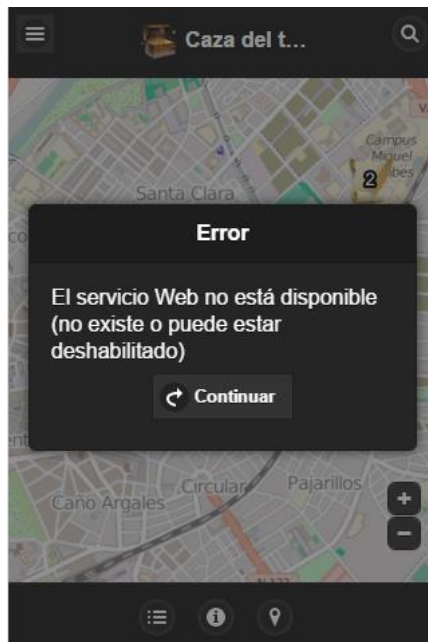


Figura 5.35 Notificación de error.

## 6. Conclusiones y líneas futuras

En este apartado se tratarán las conclusiones obtenidas con la realización de este TFG, además de las posibles líneas futuras por las que se podría continuar trabajando en este proyecto.

### 6.1. Conclusiones

Este trabajo de fin de grado, ha tenido por objetivo el diseño y la implementación de un módulo docente para la plataforma Moodle que permitiera la adaptación del juego de la caza del tesoro al plano tecnológico. De esta manera, se esperaba potenciar los nuevos métodos docentes que subyacen de los avances tecnológicos y los nuevos planes de estudio.

Para lograr dicho objetivo, se han llevado a cabo una serie de procedimientos y etapas descritos en el documento que dan como resultado una aplicación con las siguientes características:

- Una interfaz de edición simple, elegante y cómoda para el usuario final que incluye un bloqueo de edición para la integridad de los cambios realizados.
- Una interfaz de juego interactiva, estéticamente cuidada y totalmente adaptada a dispositivos móviles, sin perder la capacidad de acceso desde ordenadores ni el sincronismo del progreso del usuario con el servidor.
- Una amplia gama de parámetros de configuración de la actividad para poder recrear todos los casos de juego posible; como el modo individual o en grupo, el método de juego en movimiento o con bandera virtual y el soporte de los distintos sistemas de calificación automática.
- Posibilidad de distintas configuraciones en el diseño de cada etapa, pudiendo incluir contenido multimedia y permitiendo añadir restricciones como preguntas simples o actividades del curso a completar para su superación.
- Capacidad de visualización general del progreso de los usuarios y de los intentos realizados por cada uno de ellos en tiempo real por parte del profesor.
- Cumplimiento de todas las restricciones y requisitos de una caja del tesoro.

Con lo expuesto hasta el momento, se han satisfecho el 100% de los objetivos fijados al inicio del proyecto, además de haber cumplido las siguientes competencias asociadas al grado:

- Aprendizaje autónomo: utilizando los medios y la documentación al alcance se han conseguido resolver los problemas que iban surgiendo aportando la mejor solución de entre las posibles.
- Comunicación de ideas con claridad y concisión, necesarias para la elaboración de esta memoria.

Las implicaciones que tiene este trabajo sobre el mundo académico pueden cambiar la mentalidad generalizada de que la enseñanza se imparte dentro de las aulas. Las posibilidades que ofrece el módulo docente de la caza del tesoro pretende abrir los ojos a los profesores y animarles a crear oportunidades de aprendizaje para sus estudiantes más allá de los límites del aula tradicional. Para conseguir el mayor impacto posible, el directorio del módulo se ha subido a la plataforma GitHub y es accesible desde [31].

Por último, y desde un punto de vista personal, la realización de este trabajo ha supuesto el reto más importante de toda la carrera. Desde un principio, se ha afrontado este proyecto como un medio para demostrar la capacidad de convertir las ideas en hechos. La responsabilidad de la toma de decisiones siempre ha hecho aflorar la duda de si era la mejor elección de entre todas las posibles, consiguiendo con el paso del tiempo una mejor comprensión de los problemas y una mayor capacidad de análisis que en el inicio de esta aventura.

## 6.2. Líneas futuras

El futuro próximo del módulo pasa por su aceptación y acogida en el mundo docente como una herramienta potente para la creación de actividades georreferenciadas al aire libre. Es necesario que se incorpore progresivamente en las aulas para ver sus capacidades como nuevo método didáctico.

Aunque el módulo desarrollado para Moodle es plenamente funcional y cumple con los requisitos fijados en el inicio del proyecto, quedan en el aire una serie de prestaciones y mejoras que podrían introducirse en nuevas versiones de la aplicación:

- Implementar el uso de la tecnología QR para etapas con localizaciones interiores. Por un lado, aunque en el módulo se ha procurado garantizar el acceso a lugares cerrados, que impiden el uso de la señal GPS o que no disponen de buena cobertura móvil para la comunicación con el servidor, no es posible requerir que los alumnos estén presencialmente para descubrir la pista. Una solución a este problema, podría ser la implementación de la tecnología QR. Con un código QR impreso y colocado en la etapa a descubrir, los alumnos podrían demostrar que han estado en ese lugar para poder escanearlo.
- Adaptar el módulo de la caza del tesoro a la aplicación móvil oficial de la plataforma Moodle. A partir de la versión 3.1, Moodle permite esta adaptación a partir de los *Moodle Mobile Remote Add-ons* [32]. Esta actualización, además de otorgar mayor accesibilidad y comodidad, permitiría acceder a funciones del *smartphone* a las que no es posible acceder desde una aplicación web, como habilitar el uso de la cámara para escanear un código QR.
- Adaptar el actual sistema de configuración de actividades a completar, basado en la API de “Finalización de actividad”, cuyas condiciones de superación son pocas y demasiado simples, a la API de “Disponibilidad para elementos dentro de un módulo” [33], que permite definir condiciones más estrictas en cuanto a cuándo se considera una actividad completada.

- Enviar al usuario un aviso con los metros en línea recta desde un intento de localización fallido hasta el punto más cercano de la etapa a descubrir. Esto solucionaría la frustración de un jugador por no saber a qué distancia real se encuentra de su objetivo.
- Posibilitar un sistema colaborativo de creación de la caza del tesoro, de modo que los alumnos puedan enviar sus propios caminos para ser revisados y posteriormente habilitados por un profesor.
- Adaptar el formulario de creación de preguntas de las etapas para permitir acceder al banco de preguntas del curso, de manera que las preguntas puedan reutilizarse en futuros usos.

## 7. Referencias

- [1] M. Area Moreira y J. Adell Segura, «eLearning: Enseñar y aprender en espacios virtuales. pags. 391-424» Aljibe, Málaga, 2009.
- [2] J. M. Boneu, «Plataformas abiertas de e-learning,» *Revista de Universidad y Sociedad del Conocimiento*, vol. 4, nº 1, p. 39, 2007.
- [3] R. Baden-Powell, *Scouting for boys*, Horace Cox, 1908.
- [4] D. Kohen-Vacs, M. Ronen y S. Cohen, «Mobile Treasure Hunt Games. IEEE Technical Committee on Learning Technology,14, 24-26» 2012. Disponible en: <http://www.ieeetclt.org/issues/october2012/Kohen.pdf>. Último acceso: Julio 2016.
- [5] Geocaching, «Página oficial de Geocaching» Disponible en: <https://www.geocaching.com>. Último acceso: Julio 2016.
- [6] 7scenes, «Página oficial de 7scenes» Disponible en: <http://7scenes.com/>. Último acceso: Julio 2016.
- [7] D. Spikol y M. Milrad, «Physical Activities and Playful Learning Using Mobile Games. Research and Practice in Technology Enhanced Learning 3, 275-295» 2008.
- [8] fAR-Play, «Página oficial de fAR-Play» Disponible en: [http://hypatia.cs.ualberta.ca/aarg\\_project/far-play/index.php?page=index.php](http://hypatia.cs.ualberta.ca/aarg_project/far-play/index.php?page=index.php). Último acceso: Julio 2016.
- [9] Moodle, «Página oficial de descargas de Moodle» Disponible en: <https://download.moodle.org/>. Último acceso: Julio 2016.
- [10] Moodle, «Documentación oficial de Moodle: Requisitos del servidor en la versión 2.9» Disponible en: [https://docs.moodle.org/dev/Moodle\\_2.9\\_release\\_notes#Server\\_requirements](https://docs.moodle.org/dev/Moodle_2.9_release_notes#Server_requirements). Último acceso: Julio 2016.
- [11] Apache Friends, «Página oficial del proyecto XAMPP» Disponible en: <https://www.apachefriends.org/es/index.html>. Último acceso: Julio 2016.
- [12] Oracle Corporation, «Página oficial de Netbeans IDE» Disponible en: <https://netbeans.org/>. Último acceso: Julio 2016.
- [13] D. Rethans, «Página oficial de Xdebug» Disponible en: <https://xdebug.org/>. Último acceso: Julio 2016.

- [14] GitHub, «Página oficial de GitHub» Disponible en: <https://github.com/>. Último acceso: Julio 2016.
- [15] Moodle, «Documentación oficial de Moodle: Manual de Estilo de Código» Disponible en: [https://docs.moodle.org/all/es/Manual\\_de\\_Estilo\\_de\\_C%C3%B3digo#Estilo\\_del\\_C.C3.B3digo](https://docs.moodle.org/all/es/Manual_de_Estilo_de_C%C3%B3digo#Estilo_del_C.C3.B3digo). Último acceso: Julio 2016.
- [16] Moodle, «Documentación oficial de Moodle.Getting started. Base module folder content» Disponible en: [https://docs.moodle.org/dev/NEWMODULE\\_Documentation](https://docs.moodle.org/dev/NEWMODULE_Documentation). Último acceso: Julio 2016.
- [17] MapFish, «MapFish framework» Disponible en: <http://www.mapfish.org/index.html>. Último acceso: Julio 2016.
- [18] GeoJSON, «Especificación del formato GeoJSON» Disponible en: <http://geojson.org/geojson-spec.html>. Último acceso: Julio 2016.
- [19] OGC and ISO, «Definición del formato WKT» Disponible en: <http://docs.opengeospatial.org/is/12-063r5/12-063r5.html>. Último acceso: Julio 2016.
- [20] RequireJS, «Why AMD» Disponible en: <http://requirejs.org/docs/whyamd.html>. Último acceso: Julio 2016.
- [21] RequireJS, «Cargador de ficheros y modulos JavaScript.» Disponible en: <http://requirejs.org/>. Último acceso: Julio 2016.
- [22] Grunt, «Automatizador de tareas» Disponible en: <http://gruntjs.com/>. Último acceso: Julio 2016.
- [23] B. Morrison, «Repositorio oficial de la librería JavaScript Geocoder» Disponible en: <https://github.com/geocoder-php/geocoder-js>. Último acceso: Julio 2016.
- [24] jQuery Mobile, «Página oficial de jQuery Mobile: framework para el diseño de aplicaciones webs accesibles desde dispositivos móviles.» Disponible en: <https://jquerymobile.com/>. Último acceso: Julio 2016.
- [25] D. Furfero, «Página oficial de jQuery UI Touch Punch: código para posibilitar la interfaz jQuery UI en dispositivos táctiles.» Disponible en: <http://touchpunch.furf.com/>. Último acceso: Julio 2016.
- [26] Openlayers, «Página oficial de OpenLayers 3: librería JavaScript para la creación de mapas interactivos.» Disponible en: <http://openlayers.org/>. Último acceso: Julio 2016.



- [27] C. Wanstrath, «Mustache: Logic-less templates.» Disponible en: <https://mustache.github.io/mustache.5.html>. Último acceso: Julio 2016.
- [28] M. Antón Rodríguez y M. Á. Pérez Juárez , «Bases de datos relacionales. Documentación de la asignatura de Tecnologías para Aplicaciones Web.» Universidad de Valladolid, 2014.
- [29] AssemblySys dataServices, «PHP point-in-polygon algorithm» Disponible en: <http://assemblysys.com/php-point-in-polygon-algorithm/>. Último acceso: Julio 2016.
- [30] Moodle, «Documentación oficial de Moodle: Instalación de plugins» Disponible en: [https://docs.moodle.org/all/es/Instalar\\_plugins](https://docs.moodle.org/all/es/Instalar_plugins). Último acceso: Julio 2016.
- [31] A. Rodríguez Fernández y J. P. De Castro, «Repositorio oficial del módulo para Moodle de la caza del tesoro» Disponible en: [https://github.com/juacas/moodle-mod\\_treasurehunt](https://github.com/juacas/moodle-mod_treasurehunt). Último acceso: Julio 2016.
- [32] Moodle, «Documentación oficial de Moodle: Mobile remote add-ons» Disponible en: [https://docs.moodle.org/dev/Moodle\\_Mobile\\_Remote\\_add-ons](https://docs.moodle.org/dev/Moodle_Mobile_Remote_add-ons). Último acceso: Julio 2016.
- [33] Moodle, «Documentación oficial de Moodle: Availability API for items within a module» Disponible en: [https://docs.moodle.org/dev/Availability\\_API\\_for\\_items\\_within\\_a\\_module](https://docs.moodle.org/dev/Availability_API_for_items_within_a_module). Último acceso: Julio 2016.
- [34] The PHP Group, «¿Qué es PHP?» Disponible en: <http://php.net/manual/en/intro-what-is.php>. Último acceso: Julio 2016.
- [35] The PHP Group, «¿Qué puede hacer PHP?» Disponible en: <http://php.net/manual/en/intro-whatcando.php>. Último acceso: Julio 2016.
- [36] Mozilla Developer Network, «¿Qué es JavaScript?» Disponible en: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction#What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction#What_is_JavaScript). Último acceso: Julio 2016.
- [37] Mozilla Developer Network, «AJAX» Disponible en: <https://developer.mozilla.org/es/docs/AJAX>. Último acceso: Julio 2016.

## 8. Anexos

### 8.1. PHP

PHP es un lenguaje de scripting que puede ser embebido en HTML. Gran parte de su sintaxis es tomada de C, Java y Perl. El objetivo de este lenguaje es permitir a los desarrolladores web obtener con rapidez páginas generadas dinámicamente [34].

PHP está enfocado fundamentalmente a la programación de scripts del lado del servidor, de modo que permite realizar acciones como recopilar datos de formularios, generar páginas con contenidos dinámicos o enviar y recibir cookies. Para que esto funcione es necesario el analizador de PHP, un servidor web y un navegador web. Se debe ejecutar el servidor con una instalación de PHP conectada y acceder al resultado mediante el navegador, viendo la página a través del servidor. También es posible utilizar scripts PHP desde la línea de comandos (donde no serían necesarios el servidor ni el navegador) y escribir aplicaciones de escritorio (aunque no sería lo más apropiado para ello) [35].

Es posible utilizar PHP en sistemas operativos como Linux, Microsoft Windows, Mac OS X, RISC OS, entre otros menos conocidos. En cuanto a servidores web, admite la mayoría, entre los que podríamos destacar Apache, IIS, lighttpd y nginx. Además, existe la posibilidad de programar por procedimientos, utilizar programación orientada a objetos o incluso combinar ambas.

PHP no sólo permite generar HTML, sino que ofrece la posibilidad de crear imágenes, ficheros PDF o películas flash generadas sobre la marcha y cualquier tipo de texto. Estos ficheros pueden ser autogenerados por PHP y guardados en el sistema de ficheros en lugar de imprimirlos por pantalla, creando una caché en el lado del servidor para contenido dinámico.

En cuanto a bases de datos, PHP tiene un soporte para una amplia variedad de estas. Es posible proporcionar a una página web acceso a una base de datos utilizando una de las extensiones específicas de bases de datos o una capa de abstracción como PDO o bien conectarse a bases de datos que permitan el estándar de Conexión Abierta a Bases de Datos. También otras bases de datos podrían utilizar cURL o sockets. Además de esto PHP cuenta con soporte para comunicarse con otros servicios utilizando protocolos como LDAP, IMAP, SNMP, HTTP, etc., y soporte para el intercambio de datos complejos de WDDX y para la instalación de objetos de java, empleándolos de forma transparente como objetos PHP.

Respecto al procesamiento de texto, PHP tiene características útiles, que incluyen las expresiones regulares compatibles con Perl y extensiones y herramientas para el acceso y análisis de documentos XML.

### 8.2. JavaScript

JavaScript es un lenguaje de programación interpretado, utilizado para hacer páginas web interactivas. Es un lenguaje multiplataforma y orientado a objetos, pequeño y ligero. Este

corre en el ordenador de la persona que visita la web, de modo que no requiere descargas constantes del sitio web, siendo soportado por la mayoría de los navegadores, entre los que podemos encontrar Firefox, Chrome y Edge. JavaScript es muy utilizado para la realización de encuestas y cuestionarios.

JavaScript cuenta con una biblioteca estándar de objetos tales como Math, Array o Date, y un conjunto de elementos básicos de lenguaje donde se incluyen operadores, estructuras de control y declaraciones. Al ser un lenguaje de programación interpretado no requiere ningún programa especial para crear código útil, sino que serviría un editor de texto como Notepad.

El JavaScript básico puede ser extendido para diferentes propósitos complementándolo con objetos adicionales. Por ejemplo, JavaScript del lado del cliente cuenta con objetos para controlar un navegador y su modelo de objeto de documento. En este caso estas extensiones permiten colocar elementos en un formulario HTML y responder a eventos de usuario como podrían ser clicks con el ratón o entradas de formularios. En el caso de JavaScript del lado del servidor, las extensiones permiten por ejemplo que la aplicación se comunique con una base de datos o realizar operaciones con archivos en un servidor.

Cabe destacar que JavaScript no es lo mismo que Java, ya que en algunos casos el nombre puede llevar a confusiones. Algunas diferencias destacables serían que JavaScript es orientada a objetos y no existe distinción entre tipos de objetos, mientras que Java está basado en clases, de modo que los objetos están divididos en clases e instancias. De esta forma las propiedades y métodos de los objetos en JavaScript pueden ser añadidas a cualquier objeto de forma dinámica, mientras que en Java esto no es posible. Otra diferencia destacable es que en JavaScript los tipos de variables no son declaradas, sino que esto se realiza también dinámicamente, pero en Java es necesario declararlos (es estático). También es interesante resaltar que mediante JavaScript no es posible escribir directamente en el disco duro, mientras que en Java sí lo es [36].

### **8.3. AJAX**

AJAX no es exactamente una tecnología por sí misma, sino un término que describe un nuevo modo de utilizar conjuntamente otras tecnologías existentes, entre las cuales se encuentran: HTML o XHTML, CSS, JavaScript, DOM, XML, XSLT y el objeto XMLHttpRequest. Utilizando la combinación de estas mediante Ajax se pueden obtener aplicaciones que se actualicen automáticamente sin tener que recargar la página, debido a que intercambia pequeñas cantidades de datos con el servidor en segundo plano. De este modo, es posible crear aplicaciones más rápidas y dinámicas ante las acciones del usuario [37].

AJAX está basado en estándares de internet, usando una combinación del objeto XMLHttpRequest para recuperar datos de un servidor web, y JavaScript/DOM para mostrar/utilizar esos datos. El funcionamiento de AJAX se ve reflejado en la siguiente figura:

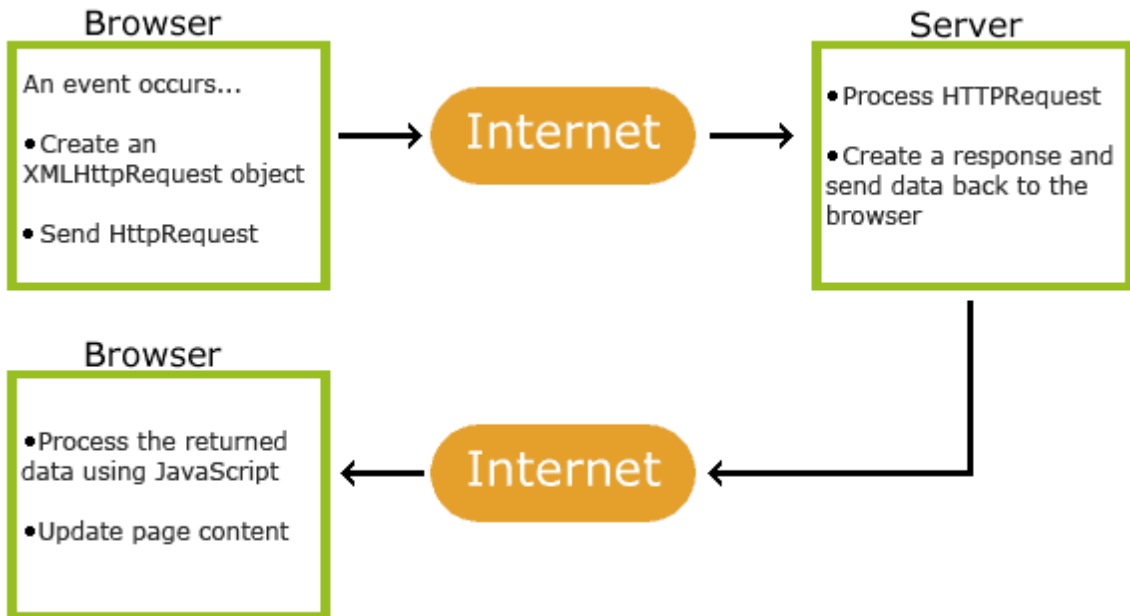


Figura 8.1 Funcionamiento de la tecnología AJAX

## 8.4. CSS

CSS es un lenguaje utilizado para definir estilos y describir cómo se van a mostrar los elementos de una página web, incluyendo el diseño, la disposición de los elementos y las variaciones en la forma de mostrarlo (por pantalla, en papel u otros medios). Se pueden utilizar diferentes estilos en función de los dispositivos utilizados y el tamaño de la pantalla.

CSS permite separar el contenido de la presentación, de modo que podamos realizar todo lo necesario en cuanto al contenido mediante HTML o XML, y utilizar CSS para definir el estilo que se empleará en ese contenido. Es posible vincular un CSS a todas las páginas web de un sitio, de modo que se pueda tener un control centralizado de la presentación facilitando de forma considerable la actualización del mismo.

CSS es un lenguaje base de la Open Web y cuenta con una especificación estandarizada por parte del W3C. Existen varias versiones: CSS1 está obsoleto actualmente, CSS2.1 es una recomendación y CSS3 es el más actual, que habitualmente es utilizado en combinación con CSS2.1.