



Universidad
Zaragoza

Trabajo Fin de Grado

Aprendiendo Jugando:

Videojuego educativo para el aprendizaje de varias
materias para alumnos de 3º ESO

Autor

Antonio Cebollada Cazallas

Director y Co-director

Director: Eduardo Mena Nieto

Co-director: José María Falcó Boudet

Escuela de Ingeniería y Arquitectura

2014

Agradecimientos

A mis directores Eduardo Mena y José María Falcó, por la oportunidad que me han ofrecido.

A mis compañeros, especialmente a Miguel Castillo y José Ignacio Castro, con los que han sido más llevaderos y amenos estos años.

A mis padres, por haberme insistido en estudiar lo que siempre me gustara, sin importar dificultad ni esfuerzos.

A Mónica, por todos estos años a mi lado apoyándome.

APRENDIENDO JUGANDO

RESUMEN

Este Trabajo Fin de Grado consiste en el diseño de un videojuego educativo para ayudar a los alumnos de secundaria a mejorar sus conocimientos en varias asignaturas. Para ello el videojuego deberá simular una aventura, donde el jugador necesitará acumular puntuación respondiendo preguntas para seguir avanzando. Durante el transcurso del juego, el alumno irá encontrando objetos que le ayuden y se irá enfrentando a distintos enemigos que impedirán su avance.

Para llevar a cabo este proyecto, se precisará diseñar un sistema que permita acceder a la aplicación a través de Internet, para que los usuarios lo utilicen desde sus propios hogares. También será necesario guardar la información importante sobre los alumnos para que el profesor pueda evaluar su progreso en el juego, por lo que se requerirá de un sistema de almacenado de la información. Debido a que todo el sistema permitirá la conexión concurrente de usuarios es necesario definir una arquitectura de distribución de la aplicación, que será de Cliente-Servidor.

Teniendo en cuenta lo anterior, el proyecto estará distribuido en los siguientes apartados:

- Introducción, donde se comentarán los ejemplos más representativos de este tipo de videojuegos, así mismo de un pequeño estudio del arte.
- Diseño de la solución, donde se abordarán los temas generales de diseño del proyecto, definiendo cada componente que se considera necesario.
- Implementación del prototipo, donde se explicarán los detalles del prototipo realizado, abordando las cuestiones más importantes de diseño y desarrollo, así mismo de las pruebas realizadas con alumnos reales y sus opiniones sobre el videojuego.
- Conclusiones, apartado en el cuál se detallará la gestión del proyecto, los problemas encontrados y también se comentarán futuras ampliaciones del videojuego, que por distintos motivos, pueden ser altamente recomendables de realizar pero que por tiempo no se han podido implementar. Como parte final, se tratarán las opiniones personales.

Índice

ÍNDICE DE FIGURAS.....	III
-------------------------------	------------

MEMORIA

1. Introducción.....	1
1.1. Gamificación. Análisis del contexto educativo.....	2
1.2. Dificultad y contexto educativo del videojuego.....	3
1.3. Estado del arte.....	5
2. Análisis y diseño de la solución.....	7
2.1. Análisis de requisitos.....	7
2.2. Arquitectura de la solución.....	8
2.3. Almacenamiento de la información.....	9
2.4. Alojamiento en un servidor Web.....	10
2.5. Sistema de soporte y ayuda.....	11
2.6. Características generales del videojuego.....	11
3. Implementación del prototipo.....	13
3.1. Tecnologías y herramientas utilizadas.....	13
3.2. Características técnicas del videojuego.....	18
3.3. Estructura de la base de datos.....	30
3.4. Funcionamiento del servidor Web	33
3.5. Pruebas realizadas.....	35
4. Conclusiones.....	37
4.1. Dedicación y gestión del proyecto.....	37
4.2. Problemas encontrados.....	42
4.3. Posibles ampliaciones.....	43
4.4. Opinión personal.....	45

BIBLIOGRAFÍA.....	47
--------------------------	-----------

Bibliografía.....	47
-------------------	----

ANEXOS

A. Estado del arte.....	51
A.1. Ejemplos de gamificación en el mercado para jóvenes.....	51
A.2. Uso de perspectiva isométrica en los videojuegos.....	53
A.3. Ejemplos de videojuegos de resolver preguntas / puzzles.....	56
B. Guía del profesor.....	59
B.1. Acceder a la aplicación.....	59
B.2. Introducción de alumnos.....	59
B.3. Introducción de preguntas.....	60
B.4. Comprobación de avances y puntuación.....	61
C. Guía del alumno.....	63
C.1. Acceder a la aplicación.....	63
C.2. Menú principal: Iniciar partida/Cargar partida.....	63
C.3. Controles básicos.....	63
C.4. Visualización del juego.....	64
D. Pruebas de servicio Web con <i>SOAPUI</i>	65
E. <i>Tiles</i> y <i>sprites</i> utilizados.....	79

Índice de Figuras

Figura 1. Diferencia del rendimiento académico con y sin uso de la gamificación en las aulas...	2
Figura 2. Videojuego Diablo II, modelo a seguir para la realización del videojuego educativo....	4
Figura 3. Diagrama general de la arquitectura.....	8
Figura 4. Verificación y descarga de los recursos de la aplicación.....	15
Figura 5. Diagrama de la arquitectura realizada.....	20
Figura 6. Cuatro perspectivas, uno desde cada punto cardinal.....	23
Figura 7. Modificación de los puntos por respuesta correcta o incorrecta.....	25
Figura 8. Restricción de cambio de nivel por poca puntuación.....	25
Figura 9. Recorrido del enemigo en busca del jugador, a través del algoritmo A*	27
Figura 10. Recorrido del jugador a una posición, a través del algoritmo A*	27
Figura 11. Interfaz de la pregunta que se muestra al jugador.....	28
Figura 12. Diagrama entidad relación de la base de datos.....	33
Figura 13. Esfuerzos realizados desglosados en cada fase.....	42
Figura 14. Duolingo, herramienta de aprendizaje de idiomas en forma de juego.....	52
Figura 15. KnowRe, herramienta educativa para aprender matemáticas.....	53
Figura 16. Imagen de juego de SimCity 2000.....	54
Figura 17. Batalla en Final Fantasy Tactics.....	55
Figura 18. Diablo 2, con imágenes 2D pero con la perspectiva isométrica.....	56
Figura 19. Videojuego Profesor Layton.....	57
Figura 20. Ejemplo de puzzle o pregunta en el Profesor Layton.....	58
Figura 21. Petición mensaje SOAP.....	66
Figura 22. Respuesta mensaje SOAP.....	66
Figura 23. Prueba de resultado correcto.....	67
Figura 24. Prueba de resultado incorrecto.....	67
Figura 25. Prueba de error.....	68
Figura 26. Sprites del héroe en reposo.....	79
Figura 27. Sprites del héroe en movimiento.....	80
Figura 28. Sprites del héroe atacando.....	80
Figura 29. Sprites héroe atacando secundario.....	81
Figura 30. Sprites héroe muriendo.....	81
Figura 31. Sprites del hombre rata en reposo.....	83
Figura 32. Sprites del hombre rata en movimiento.....	83
Figura 33. Sprites del hombre rata atacando.....	83
Figura 34. Sprites de la calavera verde en reposo.....	84
Figura 35. Sprites de la calavera verde en movimiento.....	84
Figura 36. Sprites de la calavera verde atacando.....	84
Figura 37. Sprites de la calavera blanca en reposo.....	85
Figura 38. Sprites de la calavera blanca en movimiento.....	85
Figura 39. Sprites de la calavera blanca atacando.....	85
Figura 40. Sprites del esqueleto en reposo.....	86
Figura 41. Sprites del esqueleto en movimiento.....	86
Figura 42. Sprites del esqueleto atacando.....	86
Figura 43. Sprites de la calavera negra en reposo.....	87
Figura 44. Sprites de la calavera negra en movimiento.....	87

Figura 45. Sprites de la calavera negra atacando.....	87
Figura 46. Sprites del trol de hielo en reposo.....	88
Figura 47. Sprites del trol de hielo en movimiento.....	88
Figura 48. Sprites del trol de hielo atacando.....	88
Figura 49. Sprites del trol de fuego en reposo.....	89
Figura 50. Sprites del trol de fuego en movimiento.....	89
Figura 51. Sprites del trol de fuego atacando.....	89
Figura 52. Sprites desaparición enemigo.....	90
Figura 53. Tiles de suelo y pared de los jardines (principal/secundario).....	90
Figura 54. Tiles de suelo y paredes del castillo inferior (principal/secundario).....	91
Figura 55. Tiles de suelo y paredes de la biblioteca (principal/secundario).....	91
Figura 56. Tiles de suelo y paredes del castillo superior (principal/secundario).....	91
Figura 57. Tiles de suelo y paredes del laboratorio (principal/secundario).....	92
Figura 58. Tiles de suelo y paredes de la habitación del trono (principal/secundario).....	92
Figura 59. Tiles de suelo y paredes de las torres (principal/secundario).....	92
Figura 60. Tiles de suelo y paredes de la sala del tesoro (principal/secundario).....	93
Figura 61. Tiles de suelo y paredes de la dimensión desconocida (principal/secundario).....	93
Figura 62. Sprites de los baúles.....	94
Figura 63. Sprites de las áreas de preguntas de los jardines.....	94
Figura 64. Sprites de las áreas de preguntas del castillo.....	95
Figura 65. Sprites de las áreas de preguntas de las torres.....	95
Figura 66. Sprites de las áreas de preguntas de la sala del tesoro.....	95
Figura 67. Sprites de las áreas de preguntas de la dimensión desconocida.....	96
Figura 68. Sprites de las puertas en los jardines en dirección Norte.....	97
Figura 69. Sprites de las puertas en los jardines en dirección Sur.....	97
Figura 70. Sprites de las escaleras en el castillo en dirección Norte.....	97
Figura 71. Sprites de las escaleras en el castillo en dirección Sur.....	98
Figura 72. Sprites de las escaleras en la sala del tesoro en dirección Norte.....	98
Figura 73. Sprites de las escaleras en la sala del tesoro en dirección Sur.....	98
Figura 74. Sprites de las escaleras en la dimensión desconocida en dirección Norte.....	99
Figura 75. Sprites de las escaleras en la dimensión desconocida en dirección Sur.....	99
Figura 76. Imágenes de los cursores del juego.....	100
Figura 77. Imagen de fondo de los menús iniciales del juego.....	100
Figura 78. Imagen de fondo del menú del alumno.....	101
Figura 79. Imagen de prólogo, primera parte.....	101
Figura 80. Imagen de prólogo, segunda parte.....	101
Figura 81. Imagen de prólogo, tercera parte.....	102
Figura 82. Imagen del tutorial, primera parte.....	102
Figura 83. Imagen del tutorial, segunda parte.....	102
Figura 84. Imagen del tutorial, tercera parte.....	102
Figura 85. Imagen del tutorial, cuarta parte.....	103
Figura 86. Imagen del sabio matemático.....	103
Figura 87. Imagen del sabio físico.....	103
Figura 88. Imagen del sabio químico.....	104
Figura 89. Imagen otras asignaturas.....	104
Figura 90. Imagen del antagonista del videojuego.....	104
Figura 91. Imagen de fondo de la ventana de obtención de objetos.....	105

Figura 92. Imagen de fondo del menú de fin de juego.....	105
Figura 93. Imagen de fondo del final correcto del videojuego.....	106
Figura 94. Imagen de carga inicial del juego.....	106
Figura 95. Imágenes de vida, ataque y defensa, respectivamente.....	107
Figura 96. Imágenes de las pociones.....	107
Figura 97. Imágenes del héroe.....	108
Figura 98. Imágenes del héroe, sorprendido y derrotado.....	108
Figura 99. Imágenes de los enemigos, en orden de aparición por zonas.....	108

1. Introducción

Este Trabajo Fin de Grado consiste en el diseño de un videojuego educativo que permite a los alumnos de un instituto repasar los temarios dados en clase. Funciona como una aplicación de computador, con el requisito obligado de poseer conexión a Internet. Con esto se permite que la aplicación pueda ser usada en institutos donde tengan salas destinadas a realizar clases prácticas con ordenador o desde las mismas casas de los alumnos.

Aunque el motivo principal del trabajo consiste en la aplicación del contexto educativo en un videojuego, se busca también ofrecer una buena jugabilidad que permita al alumno divertirse al mismo tiempo que consiga aprender.

Este tipo de videojuegos no han sido suficientemente explotados para un público que sufre serios problemas en la comprensión y superación de sus estudios básicos como son los adolescentes en sus asignaturas de educación secundaria. Es por esto que se ha elegido realizar este reto con el fin de promover este tipo de aplicaciones en los institutos. Como veremos más adelante, este tipo de aplicaciones permitirán a los alumnos prestar mayor atención y obtener mejores resultados mientras se entretienen con la historia del juego.

En este primer capítulo se procederá a explicar el concepto de gamificación y su relación con el proyecto, y del uso del contexto educativo en los videojuegos. Se comentará la principal dificultad a realizar con respecto a este contexto y se abordará el tema educativo presente en el proyecto y cómo se ha integrado en el mismo.

1.1. Gamificación. Análisis del contexto educativo

Se entiende como gamificación a la realización de una tarea no relacionada con los juegos a través de la mecánica y características de uno, permitiendo al usuario poner mayor atención en él, utilizando técnicas de refuerzo positivo como logros, niveles, puntuación competitiva, creando un ambiente de esfuerzo, concentración y reconocimiento personal, entre otros.

Estudios realizados por diversos centros de investigación¹ en conjunto con una universidad nacional de los *EEUU* han demostrado que la gamificación mejora considerablemente los resultados académicos de los alumnos que utilizan esas herramientas en comparación con los alumnos que no las usan. Esto lo asocian a que el alumno que utiliza una herramienta de gamificación comprende que necesita utilizar esa información educativa para progresar en el juego, y por lo tanto pone mayor interés en su estudio para poder continuar. Estos estudios también exponen que el alumno está más concentrado por el hecho de que tiene que interactuar con una herramienta que depende de sus propias acciones. Además el alumno presta mayor atención al contenido que se le proporciona, provocando que obtenga unos mejores resultados como refleja la siguiente figura.



Figura 1. Diferencia del rendimiento académico con y sin el uso de la gamificación en las aulas.

(Fuente: <http://www.edudemic.com/game-based-learning-infographic/>)

1- <http://www.edudemic.com/game-based-learning-infographic/>

Además, el uso de gamificación en el aprendizaje y la educación permite generar un ambiente de competitividad y autosuperación, con el que el alumno trabaja mejor y de forma voluntaria, buscando el reconocimiento a su trabajo a través de logros o recompensas. Según varios estudios realizados por el *MIT Education Arcade*² sólo en *EEUU* suspenden la educación secundaria y bachiller 1.3 millones de alumnos al año, comprobando que el ambiente y los métodos de evaluación y didácticos generan una desvinculación en los alumnos, que sólo se preocupan de realizar un aprendizaje memorístico para superar la formación. Sin embargo, también se ha comprobado que en el mundo entero se invierten alrededor de 3 billones de horas a la semana en jugar a videojuegos. Por este motivo, en la actualidad se intenta introducir el contexto educativo en los videojuegos, debido al gran interés que suscitan en estos momentos entre todo tipo de población, especialmente entre los jóvenes. Con toda esta información en mano, se prevé que este tipo de aplicaciones permitirán reforzar y mejorar el aprendizaje de diversas materias a los alumnos pertenecientes a la educación secundaria.

1.2. Dificultad y contexto educativo del videojuego

La principal dificultad que conlleva este proyecto es la utilización de la gamificación para un sector como es la educación secundaria, en el que no está muy extendido este tipo de aprendizaje. La gran mayoría de videojuegos educativos se centran en personas de formación inicial, como son los alumnos de educación primaria, por lo que en este sector del aprendizaje no se ha explotado y analizado lo suficiente.

Se debe recordar que ante todo, un videojuego debe ser entretenido, y por lo tanto debe equilibrar la balanza entre el aprendizaje educativo y la diversión que ofrece. Por este motivo, este proyecto intentará tener muy en cuenta estos aspectos, de forma que el alumno pueda percibir y aprender la información pedagógica de las materias mientras se sumerge en el universo del videojuego. Esta dificultad implica que el videojuego deberá presentar una jugabilidad semejante a los videojuegos que los alumnos juegan en sus hogares, impidiendo así que pierdan interés al tener otras alternativas de ocio más sugerentes.

2- <http://www.knewton.com/gamification-education/>

Una de las características más importantes que se considerarán durante el transcurso del proyecto es el de la diferenciación. Se buscará diferenciar el proyecto de otros videojuegos educativos similares, y se comprobará esta diferenciación prestando especial atención al apartado gráfico. Se procederá a desarrollar un videojuego con vista isométrica, que permite al jugador contemplar el videojuego desde una perspectiva casi 3D, o como se suele decir de los videojuegos isométricos, perspectiva 2.5D. Este tipo de perspectiva se está usando en muchos videojuegos actuales, como pueden ser las sagas *Age of Empires*, *StartCraft*, *Diablo* o *SimCity*, entre otros³.



Figura 2. Videojuego *Diablo II*, modelo a seguir para la realización del videojuego educativo.

(Fuente: <http://eu.blizzard.com/es-es/games/d2/>)

Si los videojuegos educativos no se han explotado lo suficiente, todavía menos desde una perspectiva isométrica. Por este motivo se contemplará como un buen reto la realización del proyecto en esta dirección. Para introducir de forma sencilla el contexto educativo en el videojuego, se apostará por el método de presentación de preguntas relacionadas con diversas materias, repartidas a lo largo de los distintos niveles del videojuego, de manera que el alumno obtendrá puntos si responde correctamente a las preguntas, y se le penalizará restándole puntuación si las responde mal.

3- http://en.wikipedia.org/wiki/Category:Video_games_with_isometric_graphics

1.3. Estado del arte

Como ya se ha descrito, en la actualidad existen pocos casos de gamificación cuyos usuarios objetivo sean alumnos de la educación secundaria. Esto lo podemos comprobar buscando aquellos videojuegos que se hayan comercializado o utilizado en escuelas para la enseñanza. La gran mayoría de estos videojuegos están hechos para estudiantes que están en un rango de edad muy por debajo del público objetivo del proyecto. Esto provoca que sean unos videojuegos infantiles y que nuestros usuarios no los encuentren ni atractivos ni enriquecedores, ya que la mayor parte de sus contenidos ya lo conocen y lo tienen bien asentado.

Además, el contenido que se imparte a los alumnos de la educación secundaria es más complejo de introducir en el ámbito de un videojuego, ya que se necesita la resolución de problemas y cálculos más complejos que simples operaciones básicas con las matemáticas. Como ejemplos de estos videojuegos infantiles podemos destacar los videojuegos educativos de Rayman⁴.

Todo esto proporciona la base que permitirá analizar y diseñar un videojuego adaptado a las necesidades ociosas de un usuario más ambicioso, y con un método educativo suficientemente aceptable que permite al alumno analizar, pensar y razonar sobre los problemas y las tareas que se le encomiendan en la escuela. Para un detalle más extenso sobre el estado del arte, en el que se tratan casos de gamificación en el sistema educativo y de ejemplos conocidos en temas de resolución de preguntas remitirse al Anexo A. En este anexo se trata el análisis de varios videojuegos utilizados en el contexto educativo, así mismo de varios ejemplos donde la historia del videojuego prosigue mediante la resolución de problemas o puzzles.

4- <http://www.ign.com/games/rayman-brain-games/ps-491773>

2. Análisis y diseño de la solución

En este apartado se describirán los análisis realizados para el diseño general del proyecto, comenzando por enumerar los requisitos principales del proyecto y presentando la arquitectura propuesta de la solución. Se continuará con una explicación de la necesidad de un sistema para el almacenamiento de la información y de sus registros importantes de guardar. También se introducirá la arquitectura distribuida necesaria para desarrollar el proyecto con los requisitos descritos y se comentará el sistema de soporte de la aplicación facilitado, utilizando un foro donde se expongan las dudas y problemas que puedan surgir con la aplicación. Por último, se comentarán las características generales que debe mantener cualquier videojuego que utilice este diseño.

2.1. Análisis de los requisitos

Los requisitos que deberá presentar el proyecto son los siguientes, tras haberlos definido con los directores del proyecto:

Requisitos funcionales

- RF1. Se mostrarán preguntas sobre las diversas asignaturas que deberán responder y comprender los alumnos del videojuego.
- RF2. Se mantendrá una puntuación individual para cada alumno que varíe con las acciones y respuestas del mismo.
- RF3. Existirán tres tipos de usuarios en el videojuego: Administrador, que se encarga del mantenimiento Web y de añadir a los profesores y contactar con ellos; Profesor, que introduce las preguntas en la aplicación, comprueba las evaluaciones y añade a los alumnos; y Alumno, que es el usuario que juega, responde a las preguntas y comprueba las mejores puntuaciones del videojuego.
- RF4. El usuario Profesor deberá tener acceso a la información referente a cada alumno sobre el progreso educativo en el videojuego, y el usuario Alumno deberá ser notificado del acierto o fallo de las preguntas que responde.
- RF5. El temario mostrado se centrará en las asignaturas de Matemáticas y de Física y Química del tercer curso de secundaria.

RF6. Debe poder ejecutarse desde cualquier computadora sin necesidad de estar con un navegador de Internet abierto.

RF7. El videojuego debe permitir al alumno desde interactuar con el mundo y realizar diversas acciones hasta el simple hecho de contestar preguntas.

Requisitos no funcionales

RNF1. Se requiere de una plataforma Web para alojar el servidor del videojuego.

RNF2. El idioma del videojuego es exclusivamente castellano.

RNF3. Se requiere obligatoriamente de una versión de Java superior a la 1.7.0 para poder ejecutar correctamente el videojuego.

2.2. Arquitectura de la solución

Para comprender como es la estructura de la arquitectura que permite al videojuego funcionar correctamente, se procederá a explicar cada uno de sus componentes, en alto nivel, de los que estaría formado, detallando su función dentro de la arquitectura y cómo se relaciona con los demás. El diagrama que explica su estructura es el siguiente:

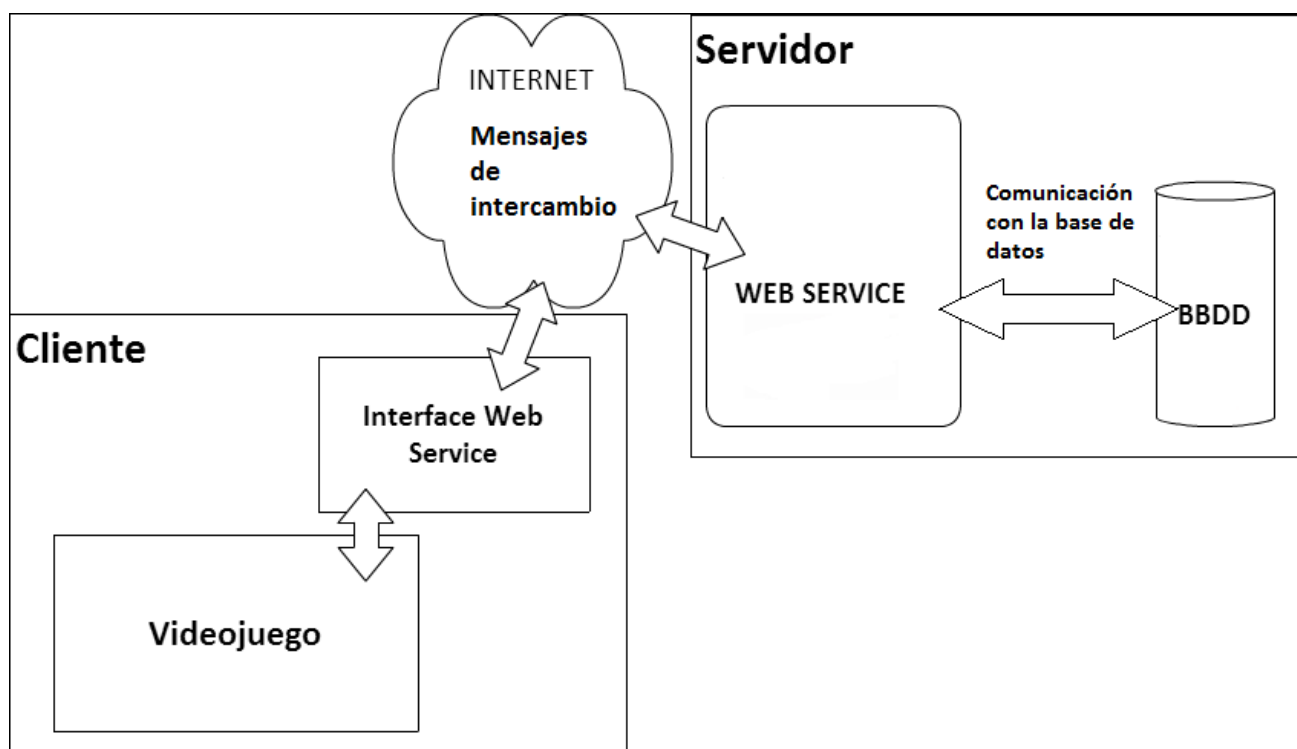


Figura 3. Diagrama general de la arquitectura.

La arquitectura se compondrá de dos elementos principales, que permitirá diferenciar la aplicación cliente y el servidor. Para realizar la comunicación del cliente con el servidor, se utilizará una estructura de llamada a servicios Web remotos a través de Internet. Este tipo de llamadas permiten comunicar al cliente con la máquina donde se aloja el servidor. Se utilizará un sistema de llamadas a Servicios Web para tratar las peticiones y respuestas que ocurren entre el cliente y el servidor, por lo que se requiere de un sistema que permita desplegar y mantener activos los servicios Web y escuchando siempre las peticiones del cliente. Este servicio se alojará en el servidor, desde donde obtendrá la información requerida por el cliente de forma local y más rápida.

Este servicio Web se comunicará internamente con un sistema que almacene la información de los usuarios, obteniéndola en el momento que lo requiera. Esta información necesitará estar relacionada entre sí para mejorar la eficiencia de las consultas, que deberán tener un tiempo de espera lo más mínimo posible para no alterar la buena experiencia del videojuego.

2.3. Almacenamiento de la información

Uno de los apartados más delicados es el elegir qué sistema de almacenamiento de información se usará para guardar la información importante del videojuego, tanto en aspectos de resultados académicos como los aspectos del propio juego. Tras analizar los posibles datos que se necesitarían guardar, se elegirá como sistema de almacenamiento una base de datos.

Debido a que se necesita tener mucha información relacionada y que de ser usada por varios institutos aumentaría considerablemente de tamaño, se comprueba que la interacción con una base de datos proporcionará más rapidez y fluidez en las consultas que se hagan sobre esta información, ya que debemos recordar que estas peticiones ocurrirán mientras el usuario está en juego, y se debe minimizar el tiempo de lectura y escritura para no repercutir en la experiencia del videojuego.

Esta opción permitirá también un mejor manejo por parte del usuario administrador, que podría realizar copias de seguridad de la base de datos en cualquier momento, para no perder información en casos de emergencia o desastre con el repositorio. También debemos tener en cuenta que este

tipo de sistema de almacenado permitirá mantener la seguridad de los datos mediante el uso de usuarios específicos para consultar y escribir en la base de datos. Esta es una de las ventajas principales ante posibles ataques al servidor, ya que el atacante no tiene los permisos para acceder al repositorio y por tanto de obtener información personal de los usuarios, que se trataría de la información más sensible en nuestro almacén.

Con todo esto, definiremos un usuario único de entrada a la base de datos, y para aumentar la seguridad al exterior este usuario será únicamente conocido por el sistema de servicios Web. Esta base de datos se explicará en detalle en la siguiente sección.

2.4. Alojamiento en un servidor Web

Debido a que el videojuego poseerá la estructura de la arquitectura Cliente-Servidor, se requerirá de un alojamiento en Web para el Servidor, y que sea accesible desde el exterior. Una vez contemos con un servidor Web será necesario instalar el sistema que gestionará los servicios Web usados por la aplicación cliente, que será la que utilizarán los usuarios finales.

El servidor Web escuchará las peticiones que la aplicación cliente le solicite, para obtenerlas y procesarlas de cara a responderle con la información requerida. Se escogerá este sistema por el principal motivo de que aumenta el grado de distribución del sistema completo, debido a que todas las aplicaciones clientes realizan peticiones contra el mismo servidor de forma concurrente y por lo tanto la base de datos es consultada desde un único punto de entrada. Esto permitirá también cerrar cualquier otra vía y ofrecer mayor seguridad al entorno.

Este servidor Web deberá también ser accesible por un usuario administrador, que pueda servir de soporte y de configuración de todo el sistema del que se compondrá el servidor. Las mejores opciones para mantener un servidor de estas características y que puedan ser totalmente accesibles por un usuario llevará a plantear varias posibilidades, como la de alojamiento en un servidor Web externo o la de creación de un propio servidor Web a través de computación en la nube, entre otros.

2.5. Sistema de soporte y ayuda

Con el fin de ofrecer una ayuda extra a los alumnos para el videojuego, se creará una página Web donde se ofrecerá la información necesaria para descargar y configurar el videojuego en el computador, además del acceso a un foro donde los usuarios podrán comentar distintos temas para mejorar la experiencia de usuario y analizar también gustos personales sobre el propio videojuego. Con este soporte se pretende también fomentar la participación activa de todos los usuarios, mejorando el contenido de la ayuda. También se ofrecerá dentro de la aplicación cliente un pequeño tutorial para explicar el funcionamiento del videojuego, tanto de control del juego como del nivel académico que conlleva.

2.6. Características generales del videojuego

Una vez tenemos el diseño general, se deben matizar aquellos aspectos más generales que deberán tenerse en cuenta a la hora de realizar un videojuego que se base en este diseño. Estos aspectos los podemos enumerar en los siguientes puntos:

1. En el videojuego deben aparecer preguntas de diversas materias que el alumno deberá responder en un determinado tiempo.
2. Se debe poder comprobar el progreso de los alumnos por parte del profesor.
3. Debe existir un *ranking* de puntuaciones en los que se registran las mejores puntuaciones de los alumnos.
4. Deben aparecer enemigos que dificulten en menor o mayor medida el avance del alumno y que puedan derrotarlo.
5. Debe existir un sistema de evolución del personaje que aumente y mejore sus habilidades.
6. El videojuego debe constar de varios niveles que serán desbloqueados progresivamente a medida que la puntuación del alumno se incrementa.
7. En el final del videojuego debe aparecer un sistema de evaluación para poder comprobar los conocimientos adquiridos por el alumno.

Todas estas características deben implementarse de la forma más conveniente en el videojuego que se elija para este sistema.

3. Implementación del prototipo

En este apartado se comentará el diseño e implementación desarrollado del prototipo del videojuego, abordando temas como las tecnologías y herramientas utilizadas para su desarrollo, la arquitectura final propuesta y explicando detalles del videojuego como pueden ser el argumento, sistema de puntuación, control del personaje y acceso a las preguntas, entre otros. En este apartado también se detallará en profundidad el funcionamiento del videojuego en un servidor Web, explicando la arquitectura utilizada y las acciones realizadas para su correcto funcionamiento.

3.1. Tecnologías y herramientas utilizadas

En este apartado se detallarán las tecnologías y herramientas de desarrollo utilizadas para el prototipo, explicando sus características y las ventajas o desventajas de las mismas.

Java

Uno de los principales motivos del uso de Java [1] en el proyecto ha sido la portabilidad y la seguridad del código, que permite ejecutar su código en cualquier computador sin la necesidad de realizar compilación de código. Otro motivo era la necesidad de realizar operaciones e instrucciones propias para diversos cálculos, como, por ejemplo, para la realización del motor gráfico. Gracias a Java se han podido realizar estas operaciones de forma mucho más compacta y legible, de manera que se optimizaban a medida que estas operaciones crecían, dividiendo el peso del cálculo en subelementos que se ejecutan de forma concurrente. Un ejemplo de esto es el tratamiento completo de los enemigos, que se ejecuta concurrentemente al hilo principal del juego, permitiendo así liberar al hilo principal de cargas y operaciones innecesarias que hubieran ralentizado y empeorado la experiencia y jugabilidad del videojuego. Además Java proporciona los recursos necesarios para realizar operaciones con la Web, a través de librerías disponibles por la comunidad y que se pueden importar al proyecto, ampliando así la potencia y funcionalidad que ya posee Java. Gracias a estos recursos, se han podido realizar las operaciones necesarias para utilizar y consumir servicios Web expuestos en un servidor Web a través de la red. Como desventaja, Java

no soporta el uso de ciertas librerías de anteriores versiones junto a otras más actuales, lo que hace que se tenga que forzar a actualizar las librerías usadas en la aplicación a las últimas versiones publicadas. Esto provoca que si un equipo con Java actualizado a la última versión ejecuta una aplicación que ha sido compilada y diseñada con librerías de una versión anterior de Java, no conseguiría hacer funcionar esa aplicación en ese equipo.

Java Web Start

Java Web Start [1] es una implementación desarrollada por *Sun Microsystems* (actualmente *Oracle*) que permite ejecutar aplicaciones Java que estén ubicadas en servidores Web, utilizando la especificación *JNLP* (*Java Network Launching Protocol*) [1]. Con esta implementación no existen los problemas y eventos que suceden en el proceso de descarga, instalación y actualización de la aplicación utilizada, ya que Java se encarga de todo de forma transparente al usuario. *Java Web Start* analiza la versión de Java instalada en el equipo cliente, y se encarga de actualizarla (o de notificar al usuario de que necesita actualizarla, si no posee los permisos necesarios para ello), para obtener siempre la versión más moderna de Java y por lo tanto, de la implementación de la aplicación. Pueden surgir problemas con las versiones de Java debido al uso de librerías de una versión específica en la aplicación, lo que impide que versiones más actuales de Java puedan utilizar dicha aplicación hasta que el administrador de la misma actualice esas librerías. Como ya hemos comentado, este tipo de librerías las actualiza Java eventualmente, impidiendo que las aplicaciones con versiones de librerías antiguas puedan ejecutarse en el nuevo entorno.

En el proyecto, se ha realizado una aplicación Java que, junto a las librerías necesarias, pueden ser ejecutadas a través de la especificación de un fichero *JNLP*. En este fichero se debe indicar la ubicación de cada uno de los ficheros necesarios para que la aplicación pueda ejecutarse, así mismo de algunas características para que pueda ejecutar la aplicación en un entorno seguro para el cliente. Esto provoca que el usuario que ejecuta el fichero *JNLP* desconozca las operaciones que *Java Web Start* realiza para descargar, instalar y ejecutar la aplicación, permitiéndole al usuario iniciar la aplicación como si se tratara de un ejecutable. Es por esto que ésta tecnología facilita en gran manera el mantenimiento y disponibilidad de la aplicación expuesta, debido a que si se ha mejorado la aplicación en el servidor con una nueva versión, *Java Web Start* comprueba que este

cambio se ha realizado y lo descarga e instala automáticamente. Gracias a esta característica, permite ofrecer siempre las mejoras de la aplicación inmediatamente y totalmente transparente al usuario.

Además, una de las características más importantes de *Java Web Start*, y con la que se diferencia por completo de las anteriores implementaciones de aplicaciones en navegador (*Applets [1]*), es que al instalarse en el equipo y guardarse en caché, se puede ejecutar sin la necesidad de un navegador. Al guardarse la información de la aplicación en caché, las próximas veces que se ejecute el archivo *JNLP* no requerirá la descarga de los recursos necesarios para la ejecución, sólo se realizaría la comprobación de que los recursos que tiene almacenados corresponden a los que existen en el servidor Web donde se expone la aplicación, como indica la siguiente figura.

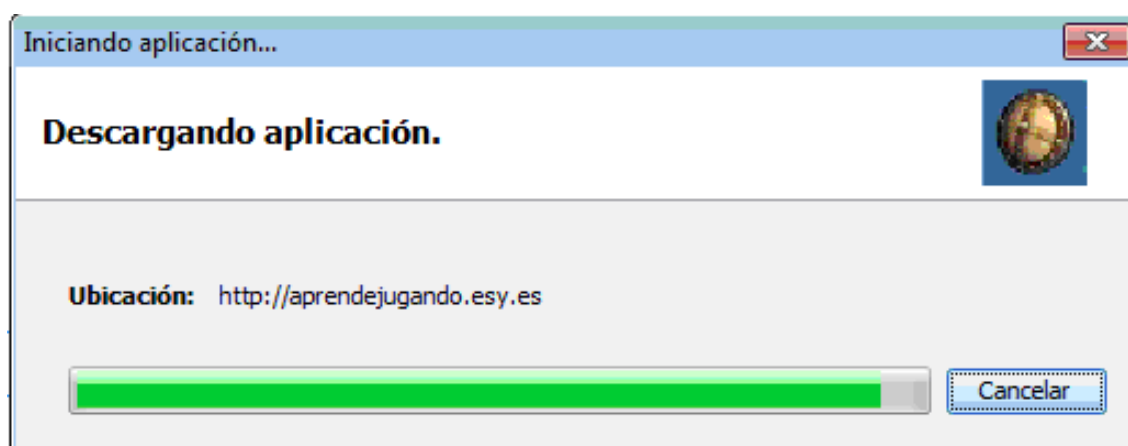


Figura 4. Verificación y descarga de los recursos de la aplicación.

Herramienta Eclipse

Para la realización del prototipo, se ha utilizado la herramienta Eclipse [2], de licencia libre, que utiliza distintos módulos para proporcionar funcionalidad al programador, permitiendo añadirle otros módulos realizados por la comunidad de usuarios.

Con un diseño muy visual, Eclipse está integrado con las herramientas de desarrollo de varios lenguajes de programación, especialmente Java, lo que le permite realizar un compilado del código automático y de proveer ayudas en los errores encontrados, siendo una característica fundamental en el desarrollo de un proyecto complejo con muchas dependencias entre clases. Como esta

herramienta dispone de una fuerte lógica que se ejecuta de forma transparente al desarrollador, permite así mismo realizar modificaciones en el código y comprobando que esos cambios no generen errores, y si los generan, dispone de la funcionalidad necesaria para modificarlo automáticamente.

Gracias a esto, Eclipse ayuda considerablemente al programador en el desarrollo, sencillez, legibilidad y robustez del código generado. Se han utilizado unas librerías externas al propio entorno Eclipse, para extender su funcionalidad. Algunas librerías utilizadas son las siguientes:

- Slick2D⁵: Librería diseñada para realizar juegos 2D en Java, ofrece muchas funcionalidades como búsqueda de caminos eficientes reproducción de animaciones y efectos especiales, entre otros. Aunque no dispone de documentación del código generado, se pueden encontrar ejemplos de uso en los foros y páginas relacionadas con esta librería. De toda la funcionalidad que dispone, sólo se ha utilizado la relativa a la búsqueda de caminos con los algoritmos A*[3], muy utilizados para obtener el mínimo recorrido a realizar entre dos puntos, conociendo los obstáculos que hay en el mapa y otorgando puntuación a cada casilla, dependiendo de si se acerca al objetivo o se aleja. Estos algoritmos son necesarios para la búsqueda de caminos a lo largo del mapa cuando un enemigo se mueve o el jugador hace clic en una zona libre del juego.
- MySQLConnector: Librería indispensable para realizar las conexiones y peticiones con una base de datos SQL, muy extendido para usarlo conjuntamente con el lenguaje Java. Esta librería permite conectarse de forma totalmente transparente al usuario y al desarrollador a una base de datos, permitiendo así generar las consultas de forma cómoda y legible.

Herramienta NetBeans

Esta herramienta se caracteriza por tener un potente y sencillo motor de diseño de pantallas, con muchos controles que permiten personalizar cada pantalla al gusto del desarrollador. NetBeans [4] también genera el código fuente de cada pantalla, permitiendo así abstraer al diseñador de del código interno a realizar para el correcto visualizado de los componentes del formulario.

5- <http://slick.ninjacave.com/>

Gracias a esta herramienta se han podido desarrollar todas las interfaces del proyecto, integrando estas pantallas en el proyecto del entorno Eclipse, comentado anteriormente. Aunque Eclipse dispone de ampliaciones que permiten generar interfaces gráficas, se ha optado por NetBeans para su desarrollo, después de analizar las características de ambos entornos.

Herramientas de diseño multimedia

Otro de los aspectos importantes del proyecto es la gran cantidad de recursos que necesita, entre ellos los correspondientes a imágenes y sonidos. Estos recursos son muy necesarios para poder dar dinamismo y acción al proyecto, debido a su naturaleza de videojuego. Las herramientas que se han usado en este proyecto para el tratamiento de los recursos se agrupan en editores de imágenes y editores de audio y música. Estas herramientas son las siguientes:

Para el tratamiento de imágenes:

- Photoshop CS6: Gran herramienta de edición, permite realizar muchas acciones sobre imágenes, como combinar varias de ellas, alterar su forma, rotarla en uno o varios ejes, modificar los colores de la imagen, etc. Este programa ha permitido generar algunos de los recursos para el proyecto, como los tiles de las paredes o suelos de algunas zonas.
- Online-Image-Editor⁶: Herramienta *Online* que permite realizar sencillas modificaciones en las imágenes, pero que posee un control muy potente y muy necesario como es el transparentar zonas específicas de la imagen, necesario para que en la visualización no se puedan superponer los colores del fondo entre imágenes.
- GraphicsMagick⁷: Programa para el sistema operativo Linux que ofrece una funcionalidad de modificación de imágenes muy potente y muy rápida, permitiendo transformar todas las imágenes deseadas en un tamaño adecuado e indicando qué colores se quieren poner como transparentes. Sin duda ha sido la mejor herramienta de edición de imágenes utilizada, debido a la rapidez en el tratamiento de numerosas imágenes a la vez, permitiendo obtener más de 50 imágenes tratadas en menos de 1 segundo.

6- <http://www.online-image-editor.com/>

7- <http://www.graphicsmagick.org/>

Para el tratamiento de sonidos y músicas:

- AudaCity⁸: Software muy potente que permite editar las músicas, amplificar su sonido, recortar partes innecesarias, añadir eco o reverberación, etc. Muy sencillo de utilizar, es el único programa utilizado para la edición de sonidos y músicas obteniendo unos resultados adecuados con cada situación.
- FormatFactory⁹ : Software que permite codificar archivos de música en distintos formatos, permitiendo transformaciones a ficheros con extensión *WAV*, *MP3*, *OGG*, etc. Sencillo de utilizar, con sólo arrastrar en la ventana de codificación los archivos deseados e indicando el formato de salida, el programa se encarga del resto. Muy necesario para reducir el peso de los ficheros de música en formato *MP3*.

3.2. Características técnicas del videojuego

El prototipo realizado es una implementación del diseño comentado anteriormente en la sección 2. Presenta unas características únicas del videojuego desarrollado, y que están sustentadas en las características generales del diseño. En este apartado se procederá a comentar la estructura del prototipo, explicando su arquitectura, los componentes que la forman y detallando cada una de sus características técnicas.

El videojuego presenta un género RPG/aventura, ambientado en la época medieval, en el que el jugador atraviesa los distintos niveles de un castillo respondiendo preguntas y eliminando enemigos, hasta llegar a la fase final donde pone a prueba sus conocimientos adquiridos en una evaluación final. Como ya se había comentado, la visualización se realiza bajo una perspectiva isométrica [5]. Debido a que los juegos medievales tienen una gran recepción por parte del tipo de usuarios al que va dirigido el proyecto, se ha escogido esta representación para el prototipo, aunque es posible llevarlo a otros ámbitos disponibles, por la flexibilidad del diseño.

8- <http://audacity.sourceforge.net/?lang=es>

9- <http://www.pcfreetime.com/>

Las características generales comentadas en la sección 2.6 se corresponden con los siguientes aspectos del videojuego implementado:

1. Las preguntas aparecen en unas zonas especiales que están distribuidas por cada nivel, en los que el jugador contacta con magos y sabios que le plantean preguntas que debe responder, y le suman puntuación si la acierta, o le resta si la falla. También, en ocasiones, el sabio químico otorga pociones de vida para recuperar salud.
2. El profesor tiene un menú con el que puede comprobar los avances de los alumnos de una clase, comprobando las preguntas acertadas, falladas, ratio de aciertos y nota de la evaluación de cada alumno, así mismo de todos los aspectos de la mejor partida registrada del mismo.
3. Cada vez que un alumno consigue una puntuación máxima, se actualiza en la base de datos un registro con esa puntuación. Los alumnos así mismo poseen un menú con el que pueden ver las puntuaciones máximas de los demás jugadores divididas por dificultad o por la clase a la que pertenece el alumno.
4. A lo largo de los niveles aparecen diversos enemigos, como hombres-rata, calaveras, esqueletos y trols, que persiguen al jugador y le atacan permitiéndole defenderse mediante un sistema de combate mediante combinaciones con el ratón para atacar.
5. A lo largo de los distintos niveles, el jugador se encuentra baúles con objetos que mejoran sus estadísticas de ataque y defensa, y también encuentra pociones de vida con las que puede curarse del daño recibido.
6. Existen 21 niveles en el videojuego, donde el jugador irá ascendiendo en el castillo y encontrándose con enemigos más fuertes y preguntas más diversas, que harán que su avance se ralentice. También están disponibles varios niveles de dificultad, Fácil, Medio y Difícil, siendo éste último el que posee retos mayores y preguntas más complicadas.
7. Una vez el alumno consigue llegar al último nivel, le espera el hechicero con una serie de veinte preguntas de las cuales deberá responder correctamente una cantidad mínima, según la dificultad de la partida, si quiere terminar satisfactoriamente el juego. Estas preguntas sirven para evaluar al alumno sobre los conocimientos adquiridos.

Estas características se detallarán posteriormente permitiendo visualizar las implementaciones realizadas para cada una de ellas. Es necesario recordar que este prototipo es solamente uno de los

muchos ejemplos que pueden realizarse con este diseño, y por ello en los apartados siguientes se mostrará la implementación específica realizada para cada característica general.

Arquitectura del prototipo

La arquitectura del prototipo desarrollado se basa en implementar la arquitectura diseñada en un principio por diversos componentes que permiten conectar todo el sistema completo. Esta arquitectura se describe gráficamente con la siguiente figura:

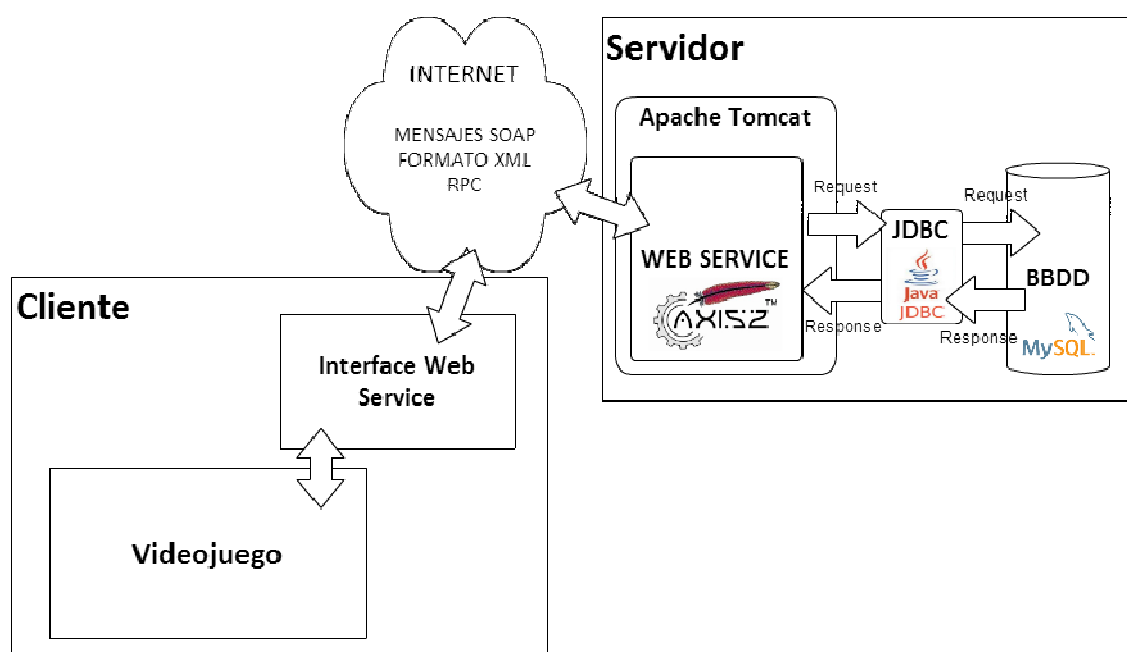


Figura 5. Diagrama de la arquitectura realizada.

La arquitectura se divide en dos componentes principales, que permiten diferenciar la aplicación cliente y el servidor. Para realizar la comunicación del cliente con el servidor, se utiliza una estructura de llamada a servicios Web mediante *RPC (Remote Procedure Call)* [1] a través de Internet. Estas llamadas permiten comunicar al cliente con el *endpoint* que corresponde al servicio Web. Para esta comunicación, se transmiten mensajes *SOAP* [1] en formato *XML*, de forma que tanto el cliente como el servidor poseen traductores *de XML* para transformar estos mensajes en las peticiones o respuestas buscadas. Una vez el cliente realiza la petición del método a través de la interfaz, se invoca remotamente el método en el servicio Web, esperando recibir la información que proporciona. El servicio Web está ejecutándose a través de un servidor Web que permite generar y desplegar de forma más sencilla y robusta estos servicios. Una vez se comunica una petición al servicio Web, éste abre una conexión con la base de datos mediante un *driver* para

realizar la consulta. Todo este sistema permite que la respuesta de la base de datos sea devuelta a la aplicación siguiendo el mismo procedimiento. Para el almacenamiento de la información, como se había descrito en el diseño, se necesitaba de una base de datos para obtener mejores resultados que ante otros sistemas de almacenado. Comprobando las distintas bases de datos disponibles, por precio, utilización y ajuste al proyecto, se ha elegido que toda la información del videojuego se almacene en una base de datos relacional de tipo *MySQL* [6]

MySQL es un sistema de gestión de bases de datos relacional, en el que las entidades de la tabla permiten tener referencias entre ellas, pudiendo así obtener mayor información con solo almacenar la clave principal de la otra tabla de la que se quiere obtener dicha información. Gracias al sistema *MySQL* se pueden utilizar bases de datos *SQL* [6], que son fácilmente accesibles a través del lenguaje Java, utilizando un *driver* para realizar las operaciones y consultas *SQL*. El *driver* utilizado para estas comunicaciones ha sido *JDBC* [1], al que sólo hay que indicar ciertos parámetros sobre el gestor *MySQL* y en qué puerto está escuchando peticiones, y automáticamente se encarga de realizar las conexiones necesarias para intercambiar mensajes. El servicio de *MySQL* necesita estar escuchando en un puerto específico para atender las peticiones que se lanzan contra las bases de datos, por lo que es importante configurar una conexión directa con el servicio que escucha en el puerto para el intercambio de mensajes. También se puede gestionar seguridad entre distintas bases de datos mediante distintos usuarios, a los que se les puede dar permisos sobre bases de datos o tablas específicas de las mismas. Con esto se gana seguridad ante ataques de obtención de información de tablas que contengan información personal de los usuarios.

Durante la realización del proyecto se ha creado un usuario especial que puede acceder a la base de datos del videojuego, siendo el único que tenga estos permisos. Debido a que esta información del mismo está alojada en el servidor y no en el cliente de la aplicación, es mucho más complicada la obtención no autorizada de información personal del resto de usuarios para acceder a la base de datos. Java también proporciona la lógica necesaria para comunicarse con una base de datos *SQL*, por lo que apoyándose en el *driver JDBC* y el sistema de gestión *MySQL*, se pueden realizar conexiones y operaciones con una base de datos de forma muy sencilla e intuitiva.

El servidor Web que se ha utilizado para mantener el servicio Web y permanecer en escucha de las peticiones de la aplicación cliente es Apache Tomcat [7], muy conocido y lo suficientemente potente para realizar la tarea descrita. Apache es un software de código libre desarrollado en Java, que proporciona una especificación para *Servlets* y *JavaServerPages (JSP)* [1]. Apache Tomcat también puede funcionar como un servidor Web por sí mismo, permitiendo desplegar y mantener aplicaciones Web.

Este software ha sido utilizado para proveer los servicios Web que consume la aplicación. Para lograrlo, se ha utilizado el sistema de extensión de Apache, comúnmente conocido como *AXIS* [8], que implementa una versión en Java y otra en C++ del protocolo *SOAP*. Este protocolo permite la interacción entre dos objetos que se encuentren en distintos procesos mediante mensajes *XML*. Debido a que se necesita contactar con el servidor para realizar operaciones con la base de datos desde la aplicación cliente, esta implementación resulta ser la solución más sencilla y eficaz para resolver este problema.

Características técnicas del videojuego

El prototipo realizado posee unas características técnicas especiales que son necesarias explicar, principalmente para comprobar cómo se han implementado y desarrollado ciertos aspectos del videojuego.

1. Argumento

El argumento del videojuego trata sobre una aventura en un castillo con el objetivo de expulsar del mismo a un enemigo poderoso. Para lograrlo, el jugador deberá ascender las salas del castillo hasta llegar al adversario, y vencerle en su mejor campo, en el del conocimiento. Durante la aventura, varios personajes ayudarán al alumno proponiéndole preguntas que debe responder en un tiempo determinado, con el fin de prepararlo para el desafío final consistente en una evaluación de muchas preguntas. Esta evaluación de preguntas determinará el final de la historia, donde el alumno puede ganar o perder.

2. Multimedia y vista isométrica

Los gráficos del videojuego consisten en una visualización con perspectiva isométrica, alternando entre los cuatro puntos cardinales de esta vista, con los que se persigue que el alumno profundice en la búsqueda de objetos o de áreas de preguntas que están ocultas. Gracias a este enfoque se consigue dar profundidad y un mayor detalle del mundo del videojuego con el que poder interactuar. Se utilizan muchas imágenes que se muestran utilizando unos cálculos matemáticos consiguiendo un buen resultado final. Para conseguir una mayor apreciación del apartado gráfico, se han utilizado una cantidad considerable de *tiles* y *sprites* y se detallan en el Anexo E. Existen aproximadamente 560 *sprites* de los personajes del juego, así mismo de 200 *tiles* de escenario y 80 imágenes de fondo y decorado diverso, siendo un total de 850 archivos de recursos gráficos, aproximadamente. Todas las interfaces están creadas bajo una apariencia medieval, de forma que se intenta integrar todas las pantallas con el estilo del videojuego.

La vista isométrica desarrollada permite al jugador visualizar el escenario desde las cuatro esquinas que se corresponden con los puntos cardinales. Gracias a esta vista se pueden realizar búsquedas de objetos y áreas de preguntas que están ocultas en el escenario. En la siguiente figura podemos distinguir las cuatro vistas de perspectiva que permite el videojuego.



Figura 6. Cuatro perspectivas, una desde cada punto cardinal, en el orden de las agujas del reloj: Oeste, Norte, Este, Sur.

La música y el sonido deben presentarse correctamente en cada situación del videojuego. Las músicas producen el efecto ambiental para cada tipo de escenario y para los distintos menús, al igual que en distintas situaciones como por ejemplo en el estado de combate, que suena una música más intensa. Los sonidos por su parte están coordinados con distintas acciones del videojuego, como abrir una puerta o un cofre, o durante el combate cada uno de los ataques de los enemigos o del propio héroe. Estos recursos proporcionan una mejor experiencia de juego al usuario, y son una parte importante en el desarrollo de un videojuego.

3. Niveles y puntuación

El videojuego posee una cantidad determinada de niveles o escenarios, cada uno distinto del anterior, que permitirán explorar al jugador por el mundo desde los jardines hasta la sala más importante del castillo, donde espera el hechicero. Estos niveles son lo suficientemente grandes para dar la sensación de perderse en el escenario del juego, además de que en cada nueva partida las escaleras y puertas para pasar a otros niveles cambian de lugar. Con esto se persigue que cada experiencia sea totalmente diferente para un jugador que ya se haya pasado el juego. Cada nivel está representado por una matriz con valores que indican qué tipo de escenario se debe colocar en cada casilla. Es por ello que todos los cálculos internos que suceden en el nivel están dirigidos contra esa matriz, permitiendo así un mayor control de todos los sucesos que ocurren en cada nivel y de una gran optimización para el cálculo para los movimientos de los enemigos. Esto permite que los movimientos realizados por el usuario en el nivel sean muy fluidos, y permiten también representar cada elemento del juego de una forma sencilla.

El sistema de puntuación es uno de los principales aspectos que permiten comprobar el avance de un alumno, ya que solo es posible obtener mucha puntuación respondiendo correctamente a las preguntas con las que el jugador se encuentre durante el juego. Con esta puntuación se controla también el avance del juego, ya que el acceso a cada nivel superior requiere un mínimo de puntuación para poder continuar. Debido a que las preguntas se pueden acertar o fallar, esto repercute aumentando o disminuyendo la puntuación total, tal y como representan las siguientes figuras de las pantallas que se muestran como resultado a la respuesta de una pregunta.

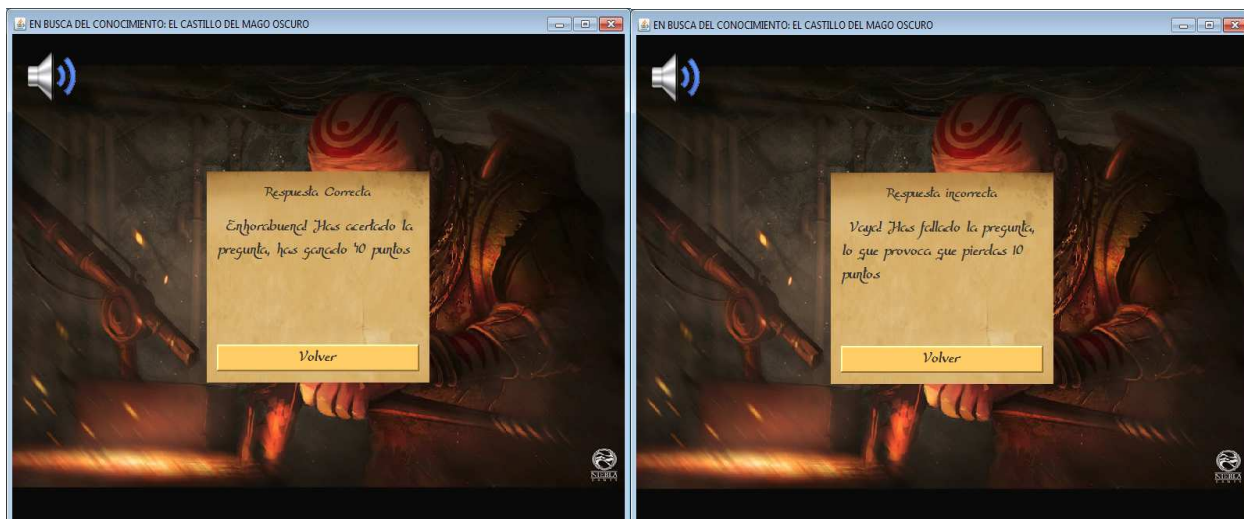


Figura 7. Modificación de los puntos por respuesta correcta o incorrecta.

Si el jugador quiere continuar al siguiente nivel pero no tiene la suficiente puntuación, se muestra una pantalla indicando el mínimo de puntos que necesita para poder continuar. Con esto se propone que el usuario permanezca más tiempo en un nivel y no pueda directamente buscar la salida para continuar en el juego. Esta pantalla que se muestra corresponde a la siguiente figura.



Figura 8. Restricción de cambio de nivel por poca puntuación.

El profesor puede comprobar toda esta puntuación obtenida del alumno, así mismo de las respuestas acertadas y falladas en cada asignatura, con lo que puede comprobar el rendimiento y mejora del alumno en el videojuego.

4. Objetos y control del personaje

En cada uno de los niveles disponibles el jugador irá encontrando cofres con objetos en su interior que son transformados en mejoras de sus estadísticas de combate, con el fin de aumentar sus posibilidades de éxito en los continuos asaltos de los enemigos que transitan por el nivel. Es obvio que a medida que se avanza en el juego los enemigos son más fuertes, por lo que la búsqueda de estos cofres es realmente necesaria para no tener problemas a medida que el jugador continúa la historia.

Un aspecto importante en un videojuego es el buen control que se tiene del personaje principal, permitiéndole realizar distintas acciones en cualquier momento en el que el usuario interactúe con él. Para poder mantener esta característica en el videojuego desarrollado se ha provisto de un control directo del héroe mediante el teclado, pero también es necesario el movimiento mediante el ratón. Un detalle muy importante a comentar es el recorrido que realiza el personaje cuando se selecciona una región del mapa con el ratón, ya que debe dirigirse correctamente hasta esa posición. Esto se consigue mediante cálculos de búsqueda optimizada usando algoritmos A* modificados para funcionar sobre este videojuego. Al estar cada uno de los niveles representado lógicamente como una matriz, estos cálculos de búsqueda funcionan perfectamente y de forma óptima para encontrar el camino adecuado en cada situación, aunque hay que realizar modificaciones especiales para el tratamiento de paredes ocultas y otros elementos que se encuentran en el mapa. El movimiento que realizan los enemigos se calcula de la misma forma que para el héroe, pero en vez de dirigirse a un lugar especificado por el jugador, se dirigen a la posición que ocupa, prevaleciendo así una sensación de persecución intensa e inteligente.

La siguiente figura muestra el camino que escogería el enemigo para encontrarse con el jugador, situado en la casilla indicada con la cruz, utilizando el cálculo de búsqueda antes descrito. El enemigo comprueba cada varios segundos la posición del jugador, para comprobar si debe realizar una nueva búsqueda de camino para llegar hasta él.



Figura 9. Recorrido del enemigo en busca del jugador, a través del cálculo de camino óptimo con el algoritmo A*.

Por otro lado, el movimiento que hace el jugador al pulsar en un lugar del mapa también genera la búsqueda del camino óptimo, tal y como se muestra en la siguiente figura, donde el jugador situado en el círculo, desea dirigirse a la posición indicada con la cruz.



Figura 10. Recorrido del jugador a una posición indicada con el ratón, a través del cálculo de camino óptimo con el algoritmo A*.

5. Áreas de preguntas

El objetivo educativo del videojuego consiste en la aparición de preguntas sobre materias que deben responder los alumnos con el fin de mejorar su capacidad cognitiva y de resolución de problemas. Estas preguntas deben contestarse en un tiempo determinado, de lo contrario se contabiliza como fallada.

El tiempo es modificable por el administrador del servidor, que lo modifica en base de datos siguiendo los consejos del profesor. Además se proporcionan pistas adicionales que deben facilitar la comprensión del enunciado o el formato requerido de la respuesta. Las preguntas también están clasificadas en distintas dificultades, que se indican cuando el profesor introduce una nueva pregunta, para poder introducir preguntas más complicadas en las dificultades más altas del videojuego y en las dificultades de juego más avanzadas. Estas preguntas se pueden responder de dos maneras, eligiendo una opción entre cuatro respuestas disponibles, o escribiendo la respuesta exacta de la pregunta. La siguiente figura muestra qué información visualiza un alumno que entre en un área de preguntas.

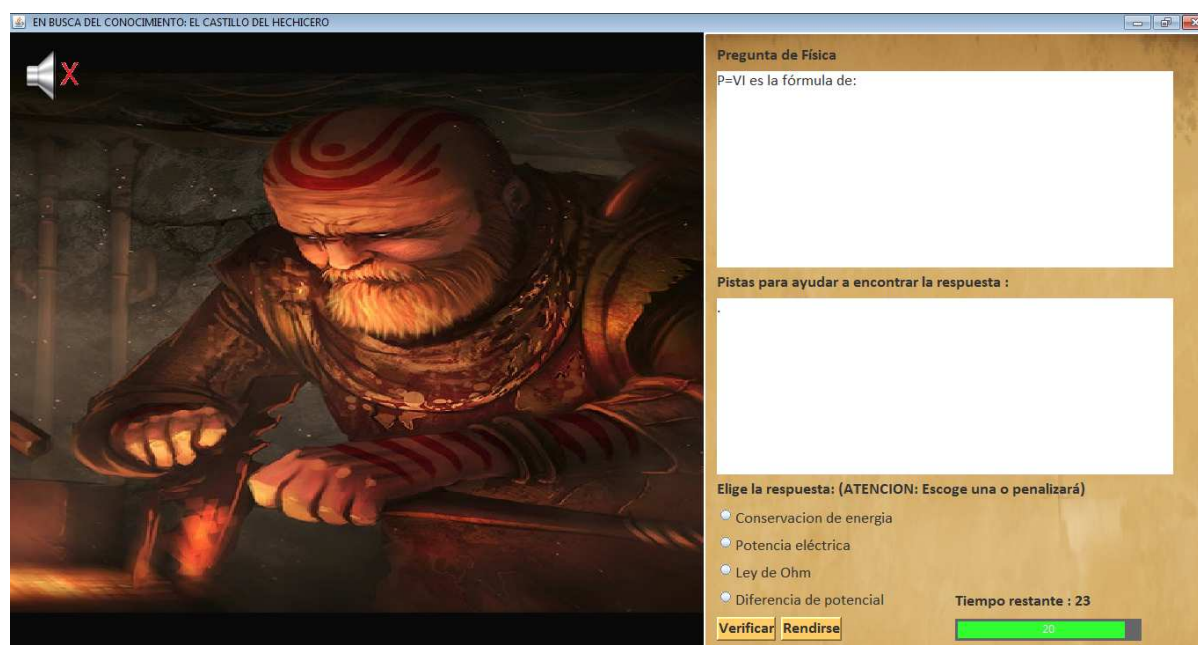


Figura 11. Interfaz de la pregunta que se muestra al jugador, con las distintas opciones disponibles e indicando el tiempo restante.

El alumno dispone de un tiempo de preparación para leer y comprender el enunciado de la pregunta, y cuando éste finaliza es cuando el alumno tiene activadas las respuestas, mientras otro tiempo de respuesta comienza a descender. También se visualiza en todo momento el tiempo disponible en pantalla, indicando con distintos colores el estado del temporizador (En verde aún falta bastante tiempo, en amarillo queda menos de la mitad, y en rojo es que quedan escasos segundos para acabar).

Un motivo importante para usar el cuadro de pistas de la pregunta es para indicar cómo debe el alumno escribir la respuesta, por el hecho de que se necesita la respuesta exactamente escrita tal y como se solicita. Esta decisión impide la aceptación de respuestas por similitud o coherencia, por lo que debe ayudar en todo lo posible al alumno para escribir la respuesta en el formato adecuado y con las necesidades requeridas de la pregunta.

Todas estas preguntas son insertadas en la aplicación por un profesor, que introduce la información necesaria para indicar qué dificultad, tipo de respuesta y tipo de asignatura corresponden a esa pregunta. En este prototipo se han insertado unas 60 preguntas de las tres asignaturas a modo de prueba, aunque el objetivo es que existan muchas más.

6. Combate e inteligencia artificial

En el prototipo los enemigos poseen una inteligencia artificial que les permite perseguir al jugador y enfrentarse a él en un combate cuerpo a cuerpo. Estos enemigos también pueden realizar acciones conjuntas, como perseguir por distintos caminos y acorralar al jugador contra una pared. Sus ataques pueden fallar o causar un daño mayor al habitual en algunas ocasiones. Para enfrentarse a estos enemigos, el jugador puede realizar unas combinaciones con la espada y escudo que terminan aturdiendo a un enemigo, permitiendo así poder realizarle un mayor daño cuando se encuentra en este estado. Además el enemigo no puede moverse ni realizar ninguna acción mientras está aturdido, lo que permite incluso la retirada del combate en el caso en el que la situación sea peligrosa.

3.3. Estructura de la base de datos

La base de datos diseñada se caracteriza por poder almacenar toda la información necesaria relativa a los conocimientos de cada alumno, como todas las variables que dependen en el juego y que permiten generar cada partida guardada con todos sus elementos. La base de datos posee varias tablas no relacionadas, con el propósito de almacenar información que pueda ser modificada directamente en base de datos. Estas últimas se explicarán al final para indicar su cometido. En este apartado se procederá a explicar cada tabla junto al diagrama entidad-relación de la base de datos diseñada. Las tablas son las siguientes:

Alumno: Esta tabla contiene la información básica del alumno, con su nombre, apellidos, clase, instituto y contraseña de acceso.

CoordenadasGuardadas: Esta tabla contiene las coordenadas de todos los objetos de un nivel en el que ha guardado la partida un alumno. Sirve para colocar cada objeto y personaje en la misma posición en la que estaba cuando se realizó el guardado.

CoordEscalerasMapa: Esta tabla guarda las coordenadas de las escaleras y del jugador en los distintos niveles que ha descubierto.

Evaluación: Esta tabla permite guardar los registros de evaluación de los jugadores, permitiendo así una obtención directa del progreso y del objetivo final del videojuego. Estas evaluaciones guardan una nota numérica que valora los conocimientos del alumno.

LogAcceso: Esta tabla está diseñada para analizar los tiempos de juego de los alumnos, y comprobar su interés al videojuego a través de los tiempos de conexión y de desconexión del mismo.

Partida: Esta tabla guarda la parte más importante del videojuego, que trata de la información de la partida guardada, como nivel, puntuación actual y número de pociones en reserva, entre otros.

Personaje: Esta tabla guarda la información de cada personaje que está ligado a una partida. En cada nueva partida se crea un personaje cuya información se actualiza con cada partida.

Pregunta: Esta tabla almacena la información de una pregunta insertada por un profesor, guardando el texto, dificultad, pistas o el tipo de pregunta, entre otros.

PuntuacionMaxima: En esta tabla se guardan los registros para cada alumno de sus mejores partidas. Con esta tabla se conoce la máxima puntuación y además la cantidad de preguntas acertadas y falladas por el alumno en esa partida.

RespondePregunta: En esta tabla se indican las respuestas dadas a las preguntas que ha realizado un alumno, indicando si la ha acertado o no. Esta tabla permite al profesor conocer cuántas preguntas ha acertado o fallado un alumno a lo largo de su dedicación al videojuego.

Respuesta: Esta tabla contiene la información de las respuestas asignadas a una pregunta.

Las tablas no relacionadas permiten almacenar información que se usa en el videojuego, y que puede ser alterada directamente desde la base de datos. Con esto se persigue que si se necesita modificar ciertos parámetros no sea necesario modificar y recompilar todo el código, de forma que es más flexible a estos cambios, y mucho más sencillo de realizarlos. Estas tablas son las siguientes:

Enemigo: Esta tabla guarda toda la información de cada enemigo que aparece en el juego. Sus valores de vida, ataque, defensa y puntuación son totalmente modificables, por lo que era necesario guardarlo en este tipo de tabla.

ErroresAplicación: Esta tabla guarda toda la información de un error ocurrido en la aplicación. Con estos errores se puede comprobar dónde y en qué método del código ha ocurrido un error, con el objetivo de mejorar la depuración del error y su seguimiento.

Profesor: Esta tabla guarda la información de un profesor para comprobar sus datos de acceso, con su nombre, apellidos, colegio y contraseña. Los profesores son directamente introducidos en la

base de datos por el administrador, permitiendo así que no se inserte un profesor no deseado desde la aplicación por parte de algún alumno.

EscaleraCambioNivel: En esta tabla se guardan las relaciones entre los distintos mapas, para poder conectarlos mediante escaleras en el nivel actual. Debido a que estos niveles son configurables, se pueden modificar los enlaces de cambio de nivel entre ellos. Por ello, los enlaces entre niveles se guardan en esta tabla.

Recursos: Esta tabla permite al administrador modificar parámetros del videojuego para añadir nuevas asignaturas o modificar la cantidad de enemigos que aparecen en un nivel, por ejemplo. Son parámetros generales sobre el juego, que no afectan a un solo jugador sino que afectan a todos, por lo que era necesario guardarlos en una tabla en la base de datos por si se quería modificar externamente.

El diagrama entidad-relación de la base de datos implementada permite identificar y cuantificar la gran cantidad de relaciones de la información almacenada en tablas que necesita un videojuego. La gran mayoría de estas relaciones son necesarias para recuperar un estado guardado de partida, y también nos permiten conocer los conocimientos de un alumno sobre las distintas preguntas que existen en el videojuego. Esta información la necesita obtener un profesor para poder evaluar al alumno en cuestión, debido a que durante el juego un alumno puede comenzar muchas partidas, responder una gran cantidad de preguntas, volver a cargar partidas anteriores, etc.

En la siguiente página se muestra el diagrama entidad-relación de la base de datos.

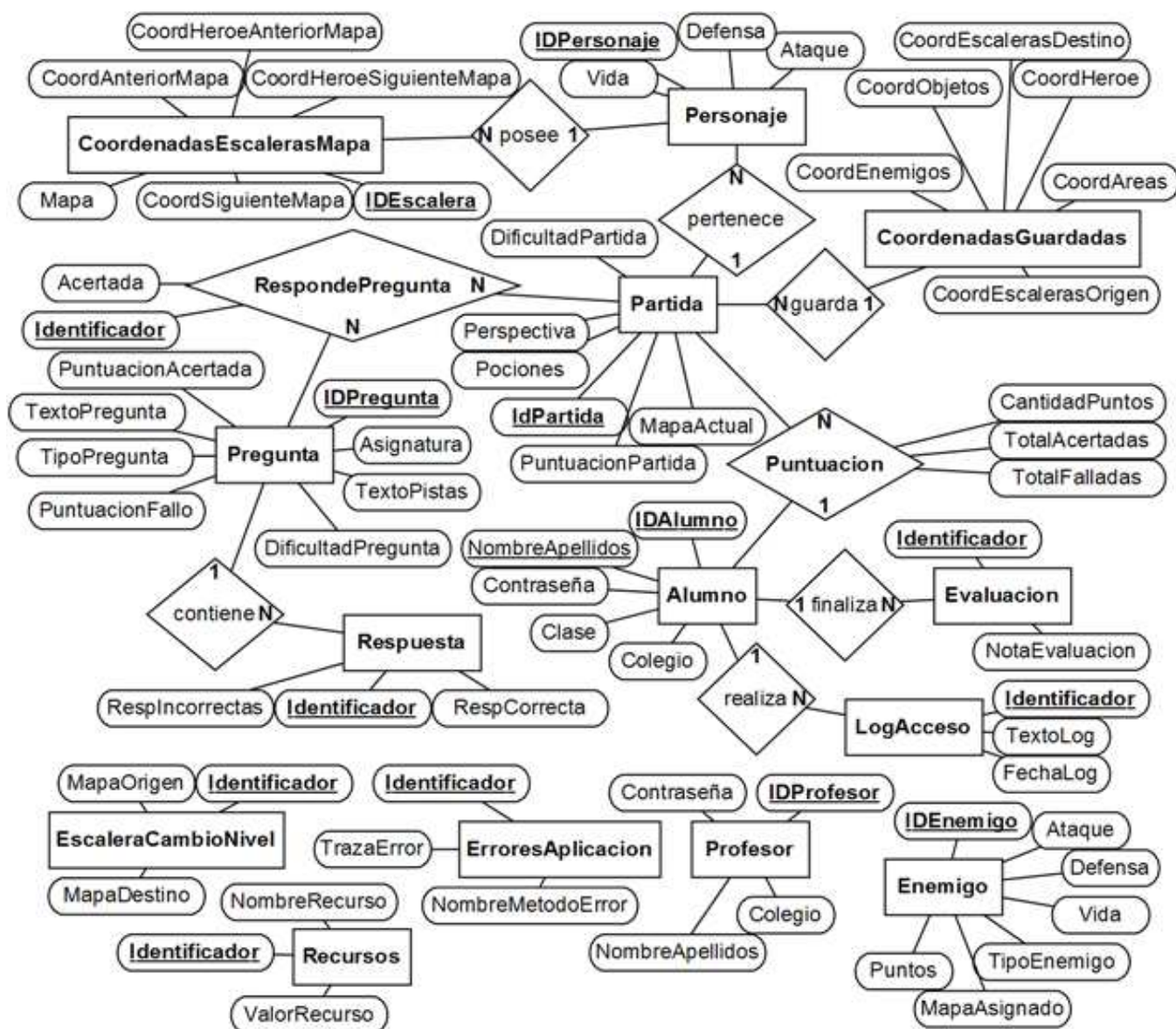


Figura 12. Diagrama entidad-relación de la base de datos.

3.4. Funcionamiento detallado del servidor Web

Como ya se ha comentado anteriormente, el servidor Web permite procesar la lógica que ocurre entre la aplicación y la información almacenada en la base de datos. Para ello, se ha especificado el uso de servicios Web siguiendo una arquitectura Cliente-Servidor, con los que la aplicación se comunica y de los que recibe la información que ha solicitado a la base de datos. Los servicios Web se han implementado utilizando servicios *SOAP* y mediante intercambio de mensajes *XML* para respuestas complejas.

Para su desarrollo se ha utilizado un software específico para mantener activo el servicio Web y se ha creado una imagen virtual donde se ubica el servicio Web y la base de datos para proveer el servicio de forma más segura. Utilizando librerías que vienen incorporadas en Java se ha permitido configurar toda esta conexión del cliente con el servicio Web. Para ello, la aplicación cliente posee varias clases encargadas de encapsular los métodos del servicio Web para proceder a su invocación en el exterior dada una URL donde está alojado el servicio Web. En la parte del servidor Web nos encontramos con una clase que contiene los métodos que acceden a la base de datos, al igual que la configuración necesaria para llevar a cabo estas operaciones con la base de datos. Esta clase que contiene los métodos expuestos del servicio Web, debe configurarse siguiendo unos pasos para poder realmente exponerlos al exterior. Para ello nos ayudamos de *AXIS* y el software *Apache Tomcat*.

Apache Tomcat permite desplegar una aplicación Web en la que se utiliza *AXIS*, y gracias a esto, se tiene accesible el servicio Web a través de la red comunicándose con el cliente mediante mensajes *SOAP* y utilizando el formato *XML* para estructurarlos. Existen dos formas de desplegar servicios Web de *AXIS* utilizando el lenguaje Java, y todas ellas generan automáticamente el fichero *WSDL* que contiene la información completa del servicio Web desplegado. La solución realizada consiste en desplegar las clases que implementan los métodos que se quieren exponer a la red mediante un descriptor de despliegue específico denominado *WSDD* [8]. Este método no implica tener el código fuente de las clases ya que trabaja únicamente con los binarios generados.

Por su parte, la aplicación consta de varias clases que se encargan de reconstruir la misma estructura que la que tiene el servicio Web, utilizando una clase de interfaz que implementará en remoto los métodos expuestos. Gracias a este sistema, la configuración queda estructurada de tal forma que se realizan las peticiones y se obtienen las respuestas del servicio Web a través de estos mensajes. Para mejorar la seguridad y robustez de la aplicación se ha desarrollado un tratamiento completo de todos los errores que pueden ocurrir, incluso de los derivados con el servicio Web o la base de datos, con el fin de que la aplicación pueda gestionar correctamente el error para indicarlo al usuario. Estos errores que pueden ocurrir en el servicio Web o en la base de datos devuelven una respuesta en modo de excepción especial de *AXIS* (*AxisFault*) con la cual se puede tratar

correctamente el error y no perder información o generar un fallo completo en la aplicación durante el intercambio de mensajes con el servidor Web.

Las pruebas realizadas para comprobar el buen funcionamiento del servicio Web se encuentran en el Anexo D. En él se describirán los métodos que expone el servicio Web, explicando con uno de ellos los tipos de pruebas realizados.

3.5. Pruebas realizadas

El prototipo se presentó ante alumnos que en la actualidad están cursando 3º de ESO en el instituto **Maria Rosa Molas**, para que aquellos interesados en probarlo facilitaran sus datos para ingresarlos en la base de datos como alumnos. En un primer momento, a unos pocos alumnos les interesó participar en este proyecto, por lo que se preparó una serie de preguntas de las asignaturas de Matemáticas, Física y Química, que suelen ser las asignaturas más complicadas en estos cursos y que mayor dificultad suponen para la mayoría de alumnos. Sin embargo, ninguno de los alumnos que se interesaron llegó a probar el videojuego.

El prototipo también fue presentado ante compañeros y amigos, que se prestaron a probarlo. Sus reacciones fueron en general muy buenas y algunos comentaron que les gustó mucho esta idea de videojuego e incluso propusieron sus ideas para mejorar el aprendizaje que se obtiene del mismo.

Los comentarios recibidos abarcan distintos tipos de opiniones, uno de ellos iba enfocado sobre el contexto educativo de las preguntas y fue el siguiente:

“El juego está muy bien pensado, pero igual debería tener unas preguntas reservadas para la evaluación que no se hayan preguntado ya. Por lo demás, el sistema me parece correcto.”

Otros comentarios sobre el videojuego en general fueron sobre la jugabilidad y dificultad del videojuego, que permitieron hacerse una idea de la propia dificultad que llevaba:

“El juego es complicado si no aciertas a atacar correctamente a los enemigos, encima aparecen muchos y no paran de seguirte. El sistema de combate podría mejorarse intentando que al marcar al enemigo se pudiera atacarle.”

“Jugar en dificultad Difícil no es un reto, es una locura, no se puede llegar ni a pasar el primer nivel...”

“Me parece muy importante la posibilidad de comprobar las puntuaciones de los demás compañeros, así se pueden realizar competiciones a ver quién llega más lejos.”

Otros comentarios también daban posibilidades de funcionalidades futuras, como se puede comprobar en el siguiente comentario:

“El videojuego está muy bien pero para ganar mucho público deberías hacerlo multijugador o, si se puede, de jugador masivo, para poder enfrentarnos también a enemigos más peligrosos mientras respondemos preguntas.”

Todos estos comentarios sobre la mejora de la jugabilidad y dificultad del videojuego son totalmente asumibles en el caso de continuar con el proyecto. Además, algunas de las mejoras propuestas son bastante aceptables para ser implementadas e incluso ser añadidas como parte del diseño.

En general, los comentarios recibidos han sido bastante positivos, y sobretodo con ideas de mejora y con ánimos para continuarlo en un futuro. Todos estos comentarios y pruebas han sido realizados por amigos, tanto de la carrera como de mi vida personal, de entre los 23 y 25 años. Añadir que no todos los compañeros que han probado el videojuego son personas que suelen jugar a videojuegos, por lo que también suponía una dificultad añadida para ellos al no conocer bien los controles, objetivos y eventos de un videojuego.

4. Conclusiones

En este apartado final se detallará la dedicación y gestión del proyecto, explicando las horas trabajadas y desglosadas en cada apartado, y las fases que se han seguido para el desarrollo. También se comentarán los principales problemas encontrados, tanto en el diseño como en el desarrollo del prototipo y las posibles ampliaciones de nuevas características que se podrían desarrollar para su integración en el proyecto. Por último, se presentarán unas opiniones personales sobre el trabajo realizado y su valoración.

4.1. Dedicación y gestión del proyecto

El prototipo se ha dividido en diversas fases para que su desarrollo fuera más sencillo, debido a que elementos diseñados en una fase se utilizarían en fases siguientes, lo que permite así basar el nuevo desarrollo en componentes robustos entre sí y aislados de forma que cada módulo diseñado se ha podido verificar por separado. Posteriormente, se verificaban en conjunto cuando ya se tenía un avance importante en el desarrollo de un módulo. Las fases realizadas, describiendo los elementos y componentes desarrollados en cada una de ellas, son las siguientes:

Diseño del motor gráfico y visualización: Esta fase ha contemplado los hitos del diseño y creación del motor gráfico inicial, analizando diseños de visualizado isométrico en juegos y adaptarlos al proyecto. Se han comprobado diversas fuentes de información que explican cuales son los pasos a seguir para realizar este tipo de motor, aunque por la existencia de diferencias con los explicados en estas fuentes de información, se ha tenido que crear una buena parte del motor gráfico implementando los cálculos y operaciones desde cero. Hasta que no se consiguió realizar un primer diseño funcional del motor gráfico, donde se visualizara el escenario con el héroe interactuando con él, no se pasó a la fase siguiente.

Diseño de la Base de Datos: Una vez se tenía un diseño aproximado de la base de datos, se dispuso a crearse la misma utilizando el gestor de bases de datos MySQL, con las dependencias entre tablas necesarias para poder funcionar. Después, se realizó una clase Java que realizara todas las

peticiones contra la base de datos, para agrupar todos los métodos que requieran esa comunicación.

Implementación de la lógica del personaje: La primera cuestión en cuanto a la lógica del personaje se focalizó en la respuesta del personaje a los distintos eventos de teclado y ratón. Primero se realizaron los eventos que ocurrían sobre el personaje cuando se utilizaba el teclado, y posteriormente se profundizó en los caminos que debía recorrer el héroe al utilizar el ratón sobre el mapa. Estos caminos realizados con algoritmos A* han sido rediseñados de los originalmente importados en la librería externa *Slick2d*, debido a que sólo permitían búsquedas en cuatro direcciones y en mapas sencillos y sin obstáculos móviles.

Correcta adecuación entre personaje, mapas y visualización: Para poder visualizar al personaje correctamente por el mapa, pareciendo que se está moviendo a lo largo del mismo, se necesitaron cálculos adicionales que comprobaban la distancia que se mueve el héroe en cada dirección. Con esto, se permite trasladar al mapa en vez del héroe, que permanece siempre en el centro de la pantalla, aunque parezca que se está moviendo por los distintos mapas del juego. También se necesitaron cálculos adicionales para comprobar si el héroe estaba oculto por alguna pared para no pintarlo, debido a que la visualización isométrica presenta una profundidad que realmente no existe.

Implementación de la lógica de las preguntas: Durante esta fase se analizó la forma de presentar al jugador el sistema de preguntas, que se había planteado como la posibilidad de realizar preguntas con texto que el jugador deba responder, ya sea mediante la respuesta escrita o eligiendo entre varias opciones disponibles. La lógica que conlleva esta presentación consiste en analizar información de la partida para obtener un tipo de pregunta, dificultad de la misma, distintas opciones de respuesta, y la puntuación que suma o resta si se acierta o falla respectivamente. Para introducir estos avances en el juego, se propuso simular el acceso a unas salas especiales donde unos personajes retaban al jugador a acertar preguntas para mejorar su puntuación.

Diseño de las interfaces de usuario: En esta fase se dedicó la mayor parte del tiempo en la realización de todos los interfaces de usuario que aparecen en el videojuego, diseñando tanto la

aparición de la pantalla como la lógica que implementa, y que permite comunicarse con otros módulos del juego. Se comenzó con las pantallas de acceso a la aplicación, donde se realizan llamadas a la base de datos para obtener información del usuario que intenta acceder y comprobar que sus datos indicados en la pantalla son correctos. También se comenzaron a realizar las pantallas de visualización de resultados de los jugadores, para ofrecer toda la funcionalidad posible al profesor que accediera a la aplicación.

Implementación de la lógica de puntuación: Uno de los problemas que surgieron una vez realizadas las fases anteriores fue el método de obtener puntuación y su función real dentro del videojuego. Se decidió la necesidad de obtener puntuación para avanzar en la historia del juego, además de poder comprobar los avances de los compañeros, y mostrando las puntuaciones obtenidas de cada uno de ellos, y su correspondiente rango dentro del juego. A medida que se obtenía puntuación o se perdía, se almacenaba junto a otra información de la partida en un objeto caché que permitía acceder mucho más rápido a la información, sin consumir servicios Web y sin colapsar la red con mensajes. También se implementó un guardado de un registro de puntuación máxima, en el que se guardaba cada mejor registro del jugador en la base de datos, para obtener las puntuaciones máximas fácilmente con varias tablas de la base de datos, y así facilitar la operación de visualización de resultados.

Implementación de la lógica de objetos, mejoras y estadísticas: Una vez se tenían los módulos principales y básicos del videojuego, se fueron realizando otros que eran secundarios para el objetivo principal del proyecto, pero que son también fundamentales en el prototipo. Se implementó la aparición de baúles con objetos que mejoraban las estadísticas del personaje para poder enfrentarse a enemigos más poderosos, que aparecerían por todo el mapa. Toda esta información sobre estadísticas, mejoras obtenidas, pociones conseguidas también se guarda en el objeto caché antes mencionado, para acceder a esta información de forma inmediata.

Implementación de la lógica de los enemigos: Esta fase fue una de las más delicadas que se habían hecho hasta el momento. Los enemigos debían aparecer por el mapa y seguir al jugador una vez se acercase al campo de visión de cada enemigo, iniciando así el sistema de combate generado para la lucha. Para poder llegar hasta el héroe desde una posición aleatoria, se introdujo lógica de búsqueda inteligente de caminos con el algoritmo A*, pero añadiendo también la inteligencia para

poder conocer la posición de otros enemigos y usarlo como ventaja. Una prueba de esto es que si un enemigo que se dirige al jugador comprueba que ya hay otro enemigo situado en alguna casilla intermedia, busca la posición que permita rodear al jugador y arrinconarlo en alguna zona del mapa. Estos enemigos también poseen la inteligencia para cambiar de dirección o de movimiento según los cambios que haga el jugador en el juego, simulando así un seguimiento completo al personaje hasta que sale del campo de visión del enemigo. En esta fase también se introdujo el sistema propio de combate, donde se realiza daño dependiendo de las estadísticas del jugador y se comprueba que la entidad situada enfrente del enemigo es el propio jugador y no otros elementos de la partida.

Recuperación del estado de partida: Una de las características que debía soportar el videojuego es el grabado de partidas para su posterior carga. Esta fase planteó problemas debido a que con la cantidad de elementos que aparecen en el mapa, se debía guardar mucha información sobre cada uno de ellos en la base de datos y de forma totalmente organizada, para acceder a ella lo más eficazmente posible. Tras varios diseños, se propuso guardar la información de la partida en cuanto a puntuación, dificultad, mapa actual, estadísticas del personaje, y todas las coordenadas que existían en el mapa, desde las de todos los personajes, como las de las escaleras, objetos y zonas de preguntas del mapa. Con esto se mantenía almacenada toda la información referente a esa partida, y así se podría reproducir la misma escena desde cero.

Implementación de la lógica de paso entre distintos niveles: Durante el transcurso del videojuego se iba accediendo a diferentes niveles, y se permitía retroceder a los ya visitados, para poder mejorar puntuación o estadísticas. Esta implementación supuso mantener todos los mapas conectados para poder viajar entre ellos y averiguar siempre en qué posición se encuentra el jugador, para así mostrarle por pantalla el nombre de la zona en la que se encuentra. También se implementaron siguiendo un patrón aleatorio todas las puertas y escaleras de cambio de nivel cuando se accede a uno nuevo, para así ofrecer una jugabilidad más extensa obligando al jugador a explorar todo el mapa cada vez que juegue. Una vez se han generado aleatoriamente, se guardan las referencias necesarias para, en la misma partida, poder volver a niveles anteriores y que esas escaleras sigan en la misma posición. Con la finalización de esta fase, se completaron todos los módulos principales del videojuego, a falta de introducir algunos ajustes secundarios que son precisamente los últimos en realizar.

Implementación del sonido y música: En todo videojuego deben presentarse músicas y sonidos que generen ambiente de fondo, así que durante esta fase se explotó esta característica, intentando optimizar su uso desde cualquier zona del código. Gracias a las librerías que ofrece Java para su reproducción no fue especialmente difícil reproducir las músicas durante el juego, aunque surgieron fallos inesperados con ciertos formatos de música, lo que replanteó utilizar otras librerías de reproducción de música, también provenientes de Java. Durante el juego se introducen también efectos sonoros de combate, o de apertura de puertas, por ejemplo, que están aislados de la reproducción de la música de fondo, con el fin de poder usar ambas de forma concurrente sin encontrar errores de reproducción típicos en la reproducción simultánea de archivos de sonido.

Mejoras en el diseño gráfico o de lógica del juego: En este último apartado se propuso abordar con todos los cambios y mejoras de las anteriores fases, una vez mostradas al profesor, para perfeccionar fallos o mejorar algunas funcionalidades que ya estaban realizadas. Algunos ejemplos fueron la posibilidad de visualizar el videojuego desde cuatro ángulos distintos, mejorando la experiencia de usuario. Este cambio fue un poco complicado de tratar, ya que no solamente bastaba con cambiar el dibujado del mapa para mostrar el tablero, sino que había que actualizar todas las referencias al mismo para que tanto enemigos, baúles, zonas de preguntas y escaleras estuvieran situadas en sus correspondientes posiciones. También se realizaron más mejoras en el apartado gráfico, y se terminó de dividir la aplicación según la arquitectura Cliente-Servidor. Con la finalización de esta fase, se dio por concluido el desarrollo del proyecto.

El esfuerzo realizado se ha contabilizado según cada tarea nueva a desarrollar, indicando las horas dedicadas al mismo, y si la tarea se ha completado con éxito.

De esta forma, se han llegado a contabilizar 780 horas de trabajo en total para la realización del proyecto, durante el transcurso de aproximadamente un año. A través de la siguiente figura se puede comprobar el esfuerzo realizado, dividido en las distintas fases ya mencionadas:

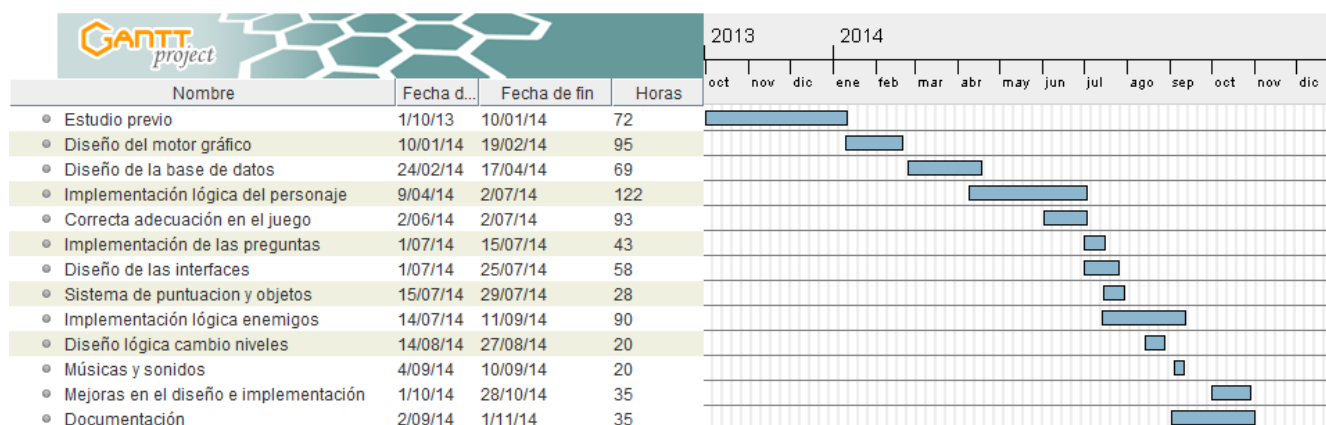


Figura 13. Esfuerzos realizados desglosados en cada tarea.

4.2. Problemas encontrados

A lo largo del desarrollo del proyecto, han surgido muchos problemas que han ido repercutiendo de una forma u otra en el tiempo invertido para desarrollar un módulo o elemento, por lo que en este apartado se comentarán aquellos problemas más representativos que han surgido.

Visualización isométrica: Uno de los problemas más importantes y recurrentes a lo largo del desarrollo del proyecto ha sido sin duda la representación visual del videojuego. La perspectiva isométrica conlleva realizar minuciosos cálculos en todo momento, ya sea para el pintado de las imágenes que forman el decorado como de la lógica de choque con objetos, paredes y enemigos. Tras varios tutoriales online y lecturas de libros sobre el tema, se consiguió realizar el motor básico que permite visualizar el conjunto de recursos en vista isométrica, y a partir de ahí se continuó.

Lógica de movimiento y acciones con el ratón: Cuando el usuario realiza una acción sobre la pantalla con el ratón, se necesitaba conocer exactamente el punto de pantalla, y su representación en la casilla correspondiente del nivel. Este tema ha dado muchos problemas ya que no se ha podido obtener un resultado perfecto en cuanto a posiciones del ratón sobre un objeto o casilla se refiere, ya que los cálculos finales permiten comprobar la casilla que correspondería con la posición del ratón, no de un objeto que hay dibujado en pantalla. No obstante, el resultado está ajustado y

es aceptable. Este problema para obtener un resultado perfecto se podría mitigar con un estudio más profundo y con más tiempo para su dedicación.

Guardar toda la información relativa a una partida en un nivel: Otro problema difícil de solucionar fue la necesidad de mantener toda la información de todos los elementos que aparecen en un nivel para poder guardarlos y luego obtenerlos en la carga de una partida. Se optó por almacenar toda la información en un objeto que permitiera conocer toda la información y modificarla en local para así cuando se realiza un guardado de partida, se pasa toda esa información al servidor que se encarga de guardarlo en la base de datos.

Mantener alto nivel de jugabilidad: Otro problema surgido fue la capacidad del videojuego para mantener al usuario lo suficientemente activo como para que no se cansara de él ni se llegue a aburrir. Para mejorar el nivel de jugabilidad, se ha desarrollado un sistema de combate inteligente, en el cual el usuario debe realizar combinaciones con el ratón para realizar ataques más poderosos a los enemigos y derrotarlos antes. También se gana en jugabilidad mostrando la información mediante estilos de interfaz más adecuadas al género del videojuego y permitiendo la recolección de objetos y pociones.

4.3. Posibles ampliaciones

A pesar del buen resultado en el trabajo realizado en el proyecto, han surgido nuevas funcionalidades que, por falta de tiempo para desarrollarlas, se han planteado como unas futuras implementaciones para el videojuego, que mejorarían la experiencia de juego y la jugabilidad del mismo. Algunas de las más importantes son las siguientes.

Habilidades de combate respondiendo preguntas: El sistema de combate podría mejorarse considerablemente si se añaden posibles movimientos extra o magias que permitan dañar a más de un enemigo, o que le provoquen estados diferentes, como sueño o miedo. Estas habilidades constarían de responder un problema práctico sobre alguna de las asignaturas en un tiempo más corto de lo normal, con el fin de que si se consigue, la habilidad se ejecuta, y sino no ocurre nada. Este sistema permitiría así al alumno pensar con más rapidez en la solución de problemas debido a la necesidad de realizar un ataque especial en un tiempo muy determinado.

Enemigos con habilidades y jefes de zona especiales: Los enemigos siguen un patrón sencillo pero efectivo para comprobar si el jugador se acerca a su campo de visión, con el objetivo de seguirle hasta alcanzarlo y golpearlo repetidas veces. Esto se podría mejorar si los enemigos también pudieran usar habilidades para retener o dañar más al jugador, o de incluso sanarse entre ellos el daño recibido. Con esto se consigue que sea más peligroso aventurarse en un grupo de enemigos juntos en vez de ir directo a por ellos sin tener en cuenta estos factores. También sería aconsejable retar al jugador en cada nueva zona con un enemigo más fuerte, o jefe especial, en el que deba resolver diversos problemas que se le plantean para ir haciendo más daño al enemigo, hasta poder vencerle para continuar.

Obtención de logros y recompensas por eventos en el juego: A parte del sistema de puntuación que permite a un alumno intentar alcanzar puestos más altos en el *ranking*, se puede introducir un sistema de logros y recompensas por la realización de diversos eventos en el juego, como pudiera ser derrotar a X enemigos o responder correctamente X preguntas, por ejemplo. Con esto también se conseguiría que los alumnos buscaran la forma de obtener todos los logros y recompensas posibles del videojuego, para poder diferenciarse de los demás por medio de estas puntuaciones extra.

Adaptación a otros idiomas: Esta es una funcionalidad imprescindible si se quiere que el videojuego traspase fronteras y se termine utilizando en otros países. Esto permitiría que este tipo de videojuegos comenzaran a expandirse por muchos lugares, de tal manera que cualquier alumno buscara aprender las asignaturas mediante un videojuego. Para esto sería necesario traducir todos los textos y preguntas que aparecen en castellano y pedir al jugador que eligiera nada más abrir la aplicación el idioma que quiere utilizar.

Mejorar las acciones con el ratón: Debido a la complejidad de los cálculos para comprobar en qué casilla está seleccionando el jugador con el ratón, el resultado no es el óptimo para obtener una jugabilidad máxima, por lo que una posible ampliación consistiría en reanalizar e implementar esta característica para que el jugador pueda seleccionar un objeto que vea y que los cálculos sean adecuados en el objeto que se esté seleccionando.

Difusión del trabajo realizado: A la hora de escribir esta memoria estamos preparando un artículo de investigación recogiendo la propuesta y experiencia de este proyecto para su publicación en alguna conferencia o revista de investigación internacional centrada en el uso de videojuegos para mejorar la docencia.

4.4. Opinión personal

Con la realización de este proyecto se ha conseguido una de las metas más buscadas de mi vida, que era el diseño y desarrollo íntegro de un videojuego para plataforma PC, además de poder realizar un prototipo con un estilo de rol fantástico, al tratarse de un género que siempre me ha atraído. Además, la opción de poder diseñar un videojuego que ayude a los más jóvenes con su educación básica era todo un reto que estuve dispuesto a tomar. Comprobé desanimado que no existían videojuegos educativos para este tipo de estudiantes, y los que había se desarrollaban igual que si fuera un videojuego para un público mucho más joven que ellos.

Una vez realizado, he podido comprobar el buen resultado del proyecto, muy contento con lo logrado pero sobretodo con lo mucho que he aprendido durante todo el proceso. He podido comprobar la gran dificultad que desentraña elaborar un diseño de un videojuego que pueda ser lo suficientemente flexible para poder implementarse con distintos tipos de ejemplos. El prototipo realizado es uno de los muchos casos que pueden implementarse utilizando el diseño del que consta este proyecto. Además, el prototipo ha entusiasmado a los usuarios que lo han probado, por lo que el resultado final, a mi juicio personal, ha sido muy bueno y sobretodo muy beneficioso para mi formación.

Para concluir, espero que este proyecto llegue a usarse realmente para la implementación de videojuegos dirigidos a alumnos de secundaria y que consiga ayudarles con sus estudios, lo cual me llenaría de alegría por ser éste el objetivo principal del proyecto.

BIBLIOGRAFÍA

- [1] **Java for Programmers** , Paul J. Deitel,Harvey M. Deitel, 2012
- [2] **Eclipse Modeling Framework: A Developer's Guide**, Frank Budinsky, David Steinberg, 2003
- [3] **Introduction to Game AI**, Neil Kirby, 2011
- [4] **The Definitive Guide to NetBean Platform 7**, Heiko Bck, 2011
- [5] **Killer Game Programming in Java**, Andrew Davison, 2005
- [6] **The Definitive Guide to MySQL 5**, Michael Kofle, 2005
- [7] **Apache Tomcat 7 Essentials**, Tanuj Khare, 2012
- [8] **Apache Axis Live: A Web Services Tutorial**, James Goodwil, 2004
- [9] **Programming Web Services With SOAP**, James Snell, 2003
- [10] **Web Services Testing with SOAPUI**, Charitha Kankanamge, 2012

