

UNIVERSIDAD DE LAS CIENCIAS
INFORMÁTICAS

Facultad 4

Análisis y diseño de la versión 1.1 de la herramienta GCon para
gestionar los contenidos de la colección La Caja Mágica

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas.

Autor:

Mayelin Castelo Rivero

Tutores:

Ing. Ana María Álvarez Valdés

Ing. Osmany Montes de Oca Rodríguez

Cotutores:

Ing. Rosalba Carralero Medina

MSc. Amilcar Rojas Taño

La Habana 2012

“Año 54 de la Revolución”

Declaración de autoría

Declaro ser la autora de la presente tesis y autorizo a la Universidad de las Ciencias Informáticas para que haga uso de la misma, como estime pendiente, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de ____ del año _____.

Mayelin Castelo Rivero

Firma del autor

Ing. Ana María Álvarez Valdés

Firma del tutor

Ing. Osmany Montes de Oca Rodríguez

Firma del tutor



*Comienzo de la sabiduría; adquiere la sabiduría; cueste lo que cueste, adquiere la
inteligencia, tenla en gran estima y ella te exaltará.*

Proverbios 4, 7- 8



Dedicatoria

Dedico la realización de este trabajo:

A Dios.

A mis padres.

A mi novio.

A mi familia.

A todos aquellos que han contribuido y hecho posible su desarrollo.



Agradecimientos

A Dios, por siempre llevarme de la mano, porque cuando me he caído él siempre ha estado ahí para ayudarme a levantar y nunca me ha desamparado.

A la madre de todos los Cubanos La virgen de la Caridad, por haber escuchado mis súplicas y haber intercedido por mi ante su hijo.

A mi mami y mi papi por su eterna confianza, por ser el motor impulsor de mi carrera, porque sin su sacrificio, apoyo, dedicación y amor incondicional no hubiera llegado hasta aquí, por no exigirme más de lo que puedo dar y siempre decirme “sí, tú puedes”, por eso quiero agradecerle todo lo que me han dado y han hecho por mí.

A mis seres más queridos, MI FAMILIA, incluyendo a María de los Ángeles y Reina la abuela de Ráidel, por ser mi sostén, ya que siempre han estado presentes, sin su apoyo y confianza no hubiera podido lograr esta meta. Especialmente a mi segunda madre, mi tía Bala, a mi abuelo Manolo y mi hermano por sus buenos consejos, su amor incondicional y saber entenderme cuando lo he necesitado.

A mi novio por su paciencia, dedicación y apoyo.

A mi abuelita querida Modesta que a pesar de no estar conmigo sé que estuviera muy orgullosa y feliz en este momento.

A las Misioneras Corazón de María por toda su ayuda espiritual y la formación que recibí de ellas desde niña, gracias a ellas y a mis padres, me he convertido en lo que soy actualmente.

A mis tíos de aquí de La Habana, por ayudarme y abrirme las puertas de su casa.

A Nairelis por ser la mejor amiga del mundo, por esos buenos momentos compartidos desde la primaria y los que nos quedan por vivir.

A mis Tutores y Co-tutores por la dura tarea de soportarme y apoyo brindado en el desarrollo de este trabajo de diploma.

A mis amigos de la UCI: Marvin, Nadiela, Lisandra, Mayara, Olia, Arlene, Yanisleidy gracias por ayudarme tanto a lo largo de mi carrera, por escuchar mis problemas, por disfrutar mis alegrías y por apoyarnos unos a los otros.

A mis compañeras de apto (Karen, Yadira, Judiht, Yisel, Linet, Celia) por estos 3 años de convivencia en los que nos supimos ayudar en las buenas y malas.

A mis compañeros de aula de todos los cursos y a todos los que conocí en determinados momentos de la carrera; ustedes compartieron momentos inolvidables de vida. A todos los llevaré presente y nunca los olvidaré.

Al colectivo de trabajadores y estudiantes del proyecto Multisaber-Navegante por el tiempo, esfuerzo y dedicación que me prestaron, ustedes estuvieron ahí siempre que los necesite y me ayudaron a formarme como ingeniera; en especial a Liana, Ismael, Licet, Osdalme, Katia, Nersa, Hector, Jorge (Patrón), Jacobo. A los estudiantes Yolanda, Ernesto, Mada, Ernesto Y, gracias por todo.

Finalmente a todas las personas que me han dedicado aunque sea un minuto de su tiempo para colaborar con el desarrollo y éxito de este trabajo, muchas gracias de todo corazón.



Resumen

En la Universidad de las Ciencias Informáticas se desarrolla una nueva versión de la colección Multisaber, esta se titula La Caja Mágica y está destinada para los niños de la enseñanza primaria de la República Bolivariana de Venezuela. La misma, presenta deficiencias en la gestión de los cuestionarios interactivos del módulo Ejercicios, para lo cual se desarrolló la herramienta GCon que los gestiona. Dicha herramienta solo abarcó la gestión de los contenidos del módulo Ejercicios, quedando pendiente la gestión de los contenidos de los juegos que requieren de información almacenada en la base de datos. Además, la herramienta no gestiona las imágenes de referencia en las tipologías de Seleccionar palabras, Completar por escritura y Completar por desplazamiento, y no permite modificar las retroalimentaciones de las dos últimas tipologías. En estas condiciones surgió la necesidad de efectuar el análisis y diseño de la versión 1.1 de la herramienta GCon para gestionar los contenidos de la colección La Caja Mágica, que no se incluyen en el panel de administración de contenido, siendo esto el objetivo del presente trabajo. Para cumplir tal tarea se utilizó la metodología de desarrollo Proceso Unificado de Desarrollo por las múltiples ventajas que ésta posee. Fueron definidos los requisitos funcionales y no funcionales, se elaboró el diagrama de casos de uso del sistema y se hizo una descripción textual de cada uno, los diagramas de clases del análisis y el del diseño también fueron creados, constituyendo el trabajo un punto de partida para la implementación de la herramienta.

Palabras clave: cuestionarios interactivos, ejercicios, Multisaber, La Caja Mágica, GCon.

Índice de contenido

Declaración de autoría	I
Dedicatoria	III
Agradecimientos.....	IV
Resumen.....	VI
Introducción	1
Capítulo 1. Fundamentación teórica	5
1.1 Introducción.....	5
1.2 Conceptos relacionados con el software educativo	5
1.2 Análisis de soluciones similares existentes.....	10
1.3 Tendencias y tecnologías actuales.....	13
1.4.1 Metodología de desarrollo	13
1.4.2 Lenguaje de modelado: Lenguaje Unificado de Modelado	16
1.4.3 Herramienta CASE: Visual Paradigm	16
1.4.4 Lenguajes del lado del cliente: JavaScript.....	17
1.4.5 Lenguajes del lado del cliente: HTML.....	18
1.4.6 Lenguajes del lado del cliente: CSS	18
1.4.7 Lenguajes del lado del servidor: PHP	19
1.4.8 Arquitectura y patrones.....	20
1.4.9 Entorno de desarrollo integrado.....	21
1.4.10 Sistemas gestores de base de datos.....	22
1.4.11 Servidores web	23
1.5 Conclusiones.....	24
Capítulo 2. Descripción de la solución propuesta	25
2.1 Introducción.....	25
2.2 Modelo de Dominio	25
2.3 Especificación de requerimientos.....	26
2.3.1 Requerimientos funcionales.....	26
2.3.2 Requerimientos no funcionales.....	31
2.4 Modelo de casos de uso del sistema	33
2.4.1 Actores del sistema	33
2.4.2 Diagrama de Casos de Uso del Sistema	33
2.4.3 Descripción de casos de uso del sistema	35
2.5 Conclusiones.....	41
Capítulo 3: Análisis y Diseño del Sistema	42
3.1 Introducción.....	42
3.2 Modelo de Análisis	42
3.2.1 Diagrama de Clases del Análisis	42
3.2.2 Diagramas de Colaboración	45
3.3 Modelo de Diseño	48

3.3.1	Patrones de Diseño	48
3.3.2	Diagrama de Clases del Diseño	49
3.4	Modelo de Datos	53
3.5	Conclusiones.....	54
	Conclusiones generales.....	55
	Recomendaciones	56
	Referencias bibliográficas	57
	Bibliografía	60
	Glosario de términos	65
	Anexos.....	67
	Anexo 1: Representación de Hipermedia.....	67
	Anexo 2: Disciplinas, fases e iteraciones de RUP.....	67
	Anexo 3: Arquitectura 3 Capas.....	68
	Anexo 4: Resumen de las descripciones de los casos de uso del sistema ..	¡Error! Marcador no definido.
	Anexo 5: Diagramas de clases del análisis	¡Error! Marcador no definido.
	Anexo 6: Diagramas de colaboración.....	¡Error! Marcador no definido.
	Anexo 7: Diagramas de clases del diseño	¡Error! Marcador no definido.

Índice de tablas

Tabla 1. Actor del sistema	33
Tabla 2. Gestionar juego de Crucigrama.....	35
Tabla 3. Gestionar juego de Adivinanza.....	36
Tabla 4. Gestionar juego de Sopa de palabras.	37
Tabla 5. Gestionar juego de Descubre la imagen 2.	38
Tabla 6. Gestionar juego de Escribiendo con el teclado.	38
Tabla 7. Gestionar juego de Escalera de palabras.....	39
Tabla 8. Gestionar tema.....	40
Tabla 9. Eliminar selección de contenidos de juego.	¡Error! Marcador no definido.
Tabla 10. Gestionar ejercicios de Seleccionar palabras.....	¡Error! Marcador no definido.
Tabla 11. Mostrar ayuda.....	¡Error! Marcador no definido.
Tabla 12. Mostrar listado de ejercicios.....	¡Error! Marcador no definido.
Tabla 13. Eliminar selección de ejercicios.....	¡Error! Marcador no definido.
Tabla 14. Mostrar cantidad de ejercicios.....	¡Error! Marcador no definido.
Tabla 15. Gestionar ejercicio de selección.....	¡Error! Marcador no definido.
Tabla 16. Gestionar ejercicio de Verdadero o falso.....	¡Error! Marcador no definido.
Tabla 17. Gestionar ejercicio de Enlazar columnas.	¡Error! Marcador no definido.
Tabla 18. Gestionar ejercicio de Formar conjuntos.....	¡Error! Marcador no definido.
Tabla 19. Gestionar ejercicios de completar.	¡Error! Marcador no definido.
Tabla 20. Gestionar ejercicio de ordenar.....	¡Error! Marcador no definido.
Tabla 21. Gestionar ejercicio de identificar.....	¡Error! Marcador no definido.
Tabla 22. Gestionar ejercicios de Seleccionar palabras.....	¡Error! Marcador no definido.
Tabla 23. Gestionar ejercicio de Seleccionar fragmento de texto. ...	¡Error! Marcador no definido.
Tabla 24. Gestionar ejercicio de Dividir palabras.	¡Error! Marcador no definido.
Tabla 25. Especificar datos generales.	¡Error! Marcador no definido.

Índice de Ilustración

Ilustración 1. Modelo del dominio.	25
Ilustración 2. Diagrama 1 de casos de uso del sistema.	33
Ilustración 3. Diagrama 2 de casos de uso del sistema.	34
Ilustración 4. Diagrama 3 de casos de uso del sistema.	34
Ilustración 5. Diagrama 4 de casos de uso del sistema.	35
Ilustración 6. Prototipo de interfaz gráfica de usuario: Juego de Crucigrama.	36
Ilustración 7. Prototipo de interfaz gráfica de usuario: Juego de Adivinanza.	37
Ilustración 8. Prototipo de interfaz gráfica de usuario: Juego de Sopa de palabras.	37
Ilustración 9. Prototipo de interfaz gráfica de usuario: Juego de Descubre la imagen 2.	38
Ilustración 10. Prototipo de interfaz gráfica de usuario: Juego de Escribiendo con el teclado.	39
Ilustración 11. Prototipo de interfaz gráfica de usuario: Juego de Escalera de palabras.	40
Ilustración 12. Prototipo de interfaz gráfica de usuario: Gestionar tema.	40
Ilustración 13. Gestionar juego Crucigrama.	43
Ilustración 14. DCA_CU Gestionar juego Adivinanza.	43
Ilustración 15. DCA_CU Gestionar juego Sopa de palabras.	44
Ilustración 16. DCA_CU Gestionar juego Descubre la imagen 2.	44
Ilustración 17. DCA_CU Gestionar juego Escribiendo con el teclado.	44
Ilustración 18. DCA_CU Gestionar juego Escalera de palabras.	45
Ilustración 19. DCA_CU Gestionar tema.	45
Ilustración 20. DC_CU Gestionar juego Crucigrama (Escenario Crear).	46
Ilustración 21. DC_CU Gestionar juego Crucigrama (Escenario Modificar).	46
Ilustración 22. DC_CU Eliminar selección de contenido de juegos.	47
Ilustración 23. DC_CU Gestionar juego Crucigrama (Escenario Eliminar un contenido de juego).	47
Ilustración 24. DCD_CU Gestionar juego Adivinanza.	50
Ilustración 25. Paquete de clases DAO de juego.	51
Ilustración 26. Paquete de clases DTO de juego.	51
Ilustración 27. Paquete de clases DAO de ejercicio.	52
Ilustración 28. Paquete de clases DTO de ejercicio.	52
Ilustración 29. Modelo de datos de ejercicios.	53
Ilustración 30. Modelo de datos de juegos.	53
Ilustración 31. Representación de hipermedia.	67
Ilustración 32. Disciplinas, fases e iteraciones de RUP.	67
Ilustración 33. Arquitectura 3 Capas.	68
Ilustración 34. DCA_CU Mostrar ayuda.	¡Error! Marcador no definido.
Ilustración 35. DCA_CU Mostrar listado de ejercicios.	¡Error! Marcador no definido.
Ilustración 36. DCA_CU Eliminar selección de ejercicios.	¡Error! Marcador no definido.
Ilustración 37. DCA_CU Mostrar cantidad de ejercicios.	¡Error! Marcador no definido.
Ilustración 38. DCA_CU Gestionar ejercicios de completar.	¡Error! Marcador no definido.

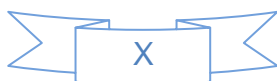


Ilustración 39. DCA_CU Gestionar ejercicios de selección.....	¡Error! Marcador no definido.
Ilustración 40. DCA_CU Gestionar ejercicios de Verdadero o falso. ¡Error! Marcador no definido.	
Ilustración 41. DCA_CU Gestionar ejercicios de Enlazar columnas. ¡Error! Marcador no definido.	
Ilustración 42. DCA_CU Gestionar ejercicios de Formar conjunto... ¡Error! Marcador no definido.	
Ilustración 43. DCA_CU Gestionar ejercicios de ordenar. ¡Error! Marcador no definido.	
Ilustración 44. DCA_CU Gestionar ejercicios de identificar. ¡Error! Marcador no definido.	
Ilustración 45. DCA_CU Gestionar ejercicios de Seleccionar palabra. ¡Error! Marcador no definido.	
Ilustración 46. DCA_CU Gestionar ejercicios de Seleccionar fragmento..... ¡Error! Marcador no definido.	
Ilustración 47. DCA_CU Gestionar ejercicios de Dividir palabras. ... ¡Error! Marcador no definido.	
Ilustración 48. DCA_CU Eliminar selección de contenidos de juegos. ¡Error! Marcador no definido.	
Ilustración 49. DC_CU Crear ejercicios de completar. ¡Error! Marcador no definido.	
Ilustración 50. DC_CU Modificar ejercicios de completar. ¡Error! Marcador no definido.	
Ilustración 51. DC_CU Eliminar selección de ejercicios..... ¡Error! Marcador no definido.	
Ilustración 52. DCD_CU Mostrar ayuda..... ¡Error! Marcador no definido.	
Ilustración 53. DCD_CU Gestionar ejercicio de completar..... ¡Error! Marcador no definido.	
Ilustración 54. DCD_CU Mostrar listado de ejercicios. ¡Error! Marcador no definido.	
Ilustración 55. DCD_CU Eliminar selección de ejercicios. ¡Error! Marcador no definido.	
Ilustración 56. DCD_CU Mostrar cantidad de ejercicios. ¡Error! Marcador no definido.	
Ilustración 57. DCD_CU Gestionar ejercicio de Seleccionar palabras..... ¡Error! Marcador no definido.	
Ilustración 58. DCD_CU Gestionar ejercicio de Enlazar columna. .. ¡Error! Marcador no definido.	
Ilustración 59. DCD_CU Gestionar ejercicio de Formar conjunto. ... ¡Error! Marcador no definido.	
Ilustración 60. DCD_CU Gestionar ejercicio de Ordenar secuencia. ¡Error! Marcador no definido.	
Ilustración 61. DCD_CU Gestionar ejercicio de selección. ¡Error! Marcador no definido.	
Ilustración 62. DCD_CU Gestionar juego Crucigrama. ¡Error! Marcador no definido.	
Ilustración 63. DCD_CU Gestionar juego Descubre la imagen 2..... ¡Error! Marcador no definido.	
Ilustración 64. DCD_CU Gestionar juego Escalera de palabras. ¡Error! Marcador no definido.	
Ilustración 65. DCD_CU Gestionar juego Escribiendo con el teclado..... ¡Error! Marcador no definido.	
Ilustración 66. DCD_CU Gestionar juego Sopa de palabras..... ¡Error! Marcador no definido.	
Ilustración 67. DCD_CU Gestionar tema. ¡Error! Marcador no definido.	

Introducción

Con el avance de las Tecnologías de la Información y las Comunicaciones (TIC) en Cuba han surgido diferentes cambios, alcanzándose un alto nivel en la producción y utilización de software. La incorporación de las nuevas tecnologías ha traído como consecuencia transformaciones en las esferas de la sociedad, fundamentalmente en la educativa. El país se encuentra inmerso en un proceso de transformación ya que los sistemas educativos de todo el mundo se enfrentan al desafío de utilizar las TIC; las cuales proporcionan a sus alumnos herramientas y conocimientos necesarios en el siglo XXI. Estos cambios en la educación hacen que los diferentes niveles de enseñanza se interesen por introducirlas en dicha esfera con el propósito de enriquecer y apoyar el proceso de enseñanza-aprendizaje.

El Ministerio de Educación (MINED) junto con un colectivo de trabajadores docentes desarrollaron varias colecciones de software para cada uno de los niveles de enseñanza del país como son: Multisaber para la enseñanza primaria, El Navegante para la secundaria y Futuro para el nivel medio superior; las cuales se encuentran instaladas en las escuelas cubanas.

En el marco de la X Mixta Cuba-Venezuela, se realizó el contrato para una nueva versión de la colección de software educativo Multisaber para la enseñanza primaria de la República Bolivariana de Venezuela. En el mismo se acordó la entrega de catorce productos para ser utilizados como herramientas de apoyo en las escuelas. A la Universidad de las Ciencias Informáticas (UCI) junto con el MINED se le asigna la tarea de desarrollar una nueva versión multiplataforma de la colección Multisaber denominada La Caja Mágica, desarrollada con herramientas libres y contextualizadas al cliente actual.

El desarrollo de la colección de software educativo La Caja Mágica viene dado por el proceso de transformación en el que se encuentra inmersa Venezuela, la necesidad de incorporar las nuevas tecnologías al sector educacional y la migración a software libre en la cual están implicadas todas sus entidades.

La Caja Mágica está integrada por varios software educativos con tecnología multimedia donde se abordan temas sobre las disciplinas Lengua Española, Matemática, Ciencias Naturales e Informática. Los elementos que la integran se gestionan a través del Sistema de Gestión de Contenidos (CMS, por sus siglas en inglés) Joomla en su versión 1.0.12; a pesar del potente panel de administración que posee, no

permite gestionar los contenidos de los cuestionarios interactivos presentes en el módulo Ejercicios, para lo cual se desarrolló la herramienta GCon que los gestiona.

La herramienta de gestión de contenidos desarrollada solo abarcó los contenidos del módulo Ejercicios, quedando pendiente los juegos, que requieren de información almacenada en la base de datos, gestionados actualmente de forma manual, interactuando directamente con esta. Además, la herramienta no gestiona las imágenes de referencia en las tipologías de Seleccionar palabras (SP), Completar por escritura (CE) y Completar por desplazamiento (CD), ni modificar las retroalimentaciones de las dos últimas tipologías.

Dada la situación problemática antes expuesta se determina como **problema a resolver** ¿Cómo gestionar los contenidos que no se incluyen en el panel de administración de contenidos de La Caja Mágica?

Para dar solución al problema planteado, se propone desarrollar la versión 1.1 de la herramienta GCon para gestionar los contenidos. En la presente investigación, por cuestiones de tiempo y escaso personal, solo se abarcará el modelado de la herramienta, indispensable para lograr la construcción del nuevo sistema.

Se define como el **objeto de estudio** el modelado de aplicaciones para la gestión de contenidos. Se establece como **campo de acción**, el modelado de aplicaciones para la gestión de contenidos de la colección La Caja Mágica.

Para dar solución al problema planteado se define como **objetivo general** diseñar la versión 1.1 de la herramienta GCon para gestionar los contenidos de la colección La Caja Mágica, con el fin de contribuir al desarrollo de una aplicación eficaz que permita gestionar los contenidos de dicha colección.

Idea a defender:

Si se realiza el análisis y diseño de la versión 1.1 de la herramienta GCon para gestionar los contenidos de la colección La Caja Mágica, entonces se contribuirá al desarrollo de una aplicación eficaz que permitirá gestionar los contenidos de dicha colección.

Del objetivo general se desglosan los siguientes **objetivos específicos**:

1. Realizar el análisis de la versión 1.1 de la GCon herramienta para gestionar los contenidos de la colección La Caja Mágica.
2. Elaborar el diseño de la versión 1.1 de la GCon herramienta para gestionar los contenidos de la colección La Caja Mágica.

Para dar cumplimiento a los objetivos expuestos se plantean las siguientes **tareas a cumplir**:

1. Realización de un estudio del estado del arte.
2. Identificación de los requisitos funcionales y no funcionales.
3. Modelación de los diagramas de casos de uso del sistema.
4. Descripción de los casos de uso del sistema.
5. Análisis de los requisitos funcionales y no funcionales.
6. Confección de los Diagramas de Clases del Análisis.
7. Confección de los Diagramas de Colaboración para los casos de uso más significativos.
8. Confección del Diagrama de Clases del Diseño.
9. Confección del Modelo de Datos.

Para un mejor desarrollo de la investigación y el futuro éxito de la misma, se utilizaron diversos **métodos de investigación**.

Métodos teóricos:

La aplicación del método **analítico – sintético** permitió el estudio y el análisis de las teorías, diferentes conceptos y definiciones, logrando la extracción de los elementos más importantes que se relacionan con el modelado de aplicaciones para la gestión de contenidos.

El método **histórico – lógico** permitió realizar un estudio de las diferentes versiones de la aplicación para gestionar los ejercicios, sus características, ventajas, así como su evolución, para poder obtener posibles mejoras a incluir en la nueva aplicación.

Método empírico:

El método empírico **observación** se utilizó con el objetivo de caracterizar la problemática definida, por lo que propició plantear el problema a resolver, definir el objeto de estudio y dentro de él, el área

determinada a trabajar.

Estructura de la tesis:

El documento consta de una introducción, tres capítulos, conclusiones, recomendaciones, sección bibliográfica y anexos; logrando una mejor distribución del contenido y claridad en las ideas. Los capítulos están estructurados de la siguiente manera:

Capítulo I: Fundamentación teórica. Se realiza un estudio del estado del arte para analizar soluciones similares a la propuesta. Se realiza la fundamentación teórica de dicha investigación, que comprende un estudio de las metodologías, herramientas y lenguajes utilizados para un correcto desarrollo del análisis y diseño de la segunda versión de una herramienta.

Capítulo II: Descripción de la solución propuesta. Se realiza el levantamiento de requisitos. Se explican las funcionalidades del sistema y se describe la solución propuesta con el uso de los diagramas de casos de uso del sistema.

Capítulo III: Análisis y diseño del sistema. Se realiza el Diagrama de clases del análisis con el propósito de refinar y estructurar los requisitos obtenidos para facilitar la comprensión, preparación, modificación y mantenimiento de los mismos. A partir del análisis se confecciona el Diagrama de clases del diseño obteniéndose una visión más detallada del sistema. Se modela la Base de Datos (BD) de la solución propuesta.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En el presente capítulo se abordan conceptos relacionados con el desarrollo de software educativo. Se hace un análisis de soluciones similares y se enfatiza en la descripción de las herramientas y tecnologías que permitirán la correcta elaboración del análisis y diseño del sistema, teniendo en cuenta sus características, ventajas y desventajas.

1.2 Conceptos relacionados con el software educativo

La necesidad que existe hoy en día de incorporar las TIC en las actividades docentes es cada vez mayor debido a que desempeñan un papel importante en este proceso, lo cual trae como consecuencia que sea superior la producción y el desarrollo de software educativos. A continuación se exponen algunas de las definiciones más comunes.

Medios de enseñanza

Los medios de enseñanza cumplen funciones instructivas, cibernéticas, formativas y recreativas, además de las funciones: motivadora-innovadora-creadora, lúdica-recreativa y desarrolladora-control, las que de manera científica, favorecen el desarrollo de la personalidad de los alumnos. Estos medios son importantes ya que permiten dar cumplimiento a los objetivos, favoreciendo que los alumnos se puedan apropiar del contenido de manera reflexiva y consciente, en una unidad entre la instrucción, la educación y el desarrollo. (1)

A partir de lo antes expuesto se puede llegar a la conclusión que los medios de enseñanza son considerados como los componentes del proceso de enseñanza-aprendizaje que sirven de sostén a los métodos determinados por el objetivo y el contenido, los que se convierten en criterios decisivos para su selección y empleo. Constituyen distintas imágenes y representaciones, abarcan objetos materiales e industriales que contienen información y se utilizan como fuentes de conocimiento que contribuyen a que la enseñanza sea activa.

Software educativo

Varios autores han definido el concepto de software educativo. A continuación se exponen algunos de estos criterios:

Se define como software educativo cualquier programa computacional cuyas características estructurales y funciones sirvan para apoyar el proceso de enseñar, aprender y administrar, es decir, un material de aprendizaje especialmente diseñado para ser utilizado en una computadora en los procesos de enseñar y aprender. (2)

Software educativo se plantea como una herramienta que proporciona un estilo de aprendizaje acorde a la realidad actual, motivando al alumno para comprender por sí mismo, hacer elecciones rápidas, razonadas y vivir creativamente, ya que le permite simular una realidad que facilita su inserción en el sistema social en el cual vive. (3)

A partir de las definiciones anteriores se considera que el software educativo es un producto informático con el que se puede aprender y está diseñado con una intencionalidad educativa, por lo que facilita impartir un conocimiento determinado al estudiante. Estos pueden tratar diferentes asignaturas, constituyendo un instrumento efectivo para el desarrollo educacional.

Multimedia

El desarrollo de las TIC, con la incorporación de las computadoras a los medios electrónicos, no deja de ser impresionante. Cada día se observan nuevas herramientas, dispositivos y técnicas que facilitan en gran medida el desarrollo de las labores diarias, como es el caso de la multimedia.

Una multimedia no es más que la integración de al menos tres de los diversos datos o información manejados por la computadora: texto, gráficas, sonido, voz y video, que hace concurrir a diversas tecnologías: de expresión, comunicación, información, sistematización y documentación, para dar lugar a aplicaciones en la educación, diversión y entretenimiento, información, comunicación, capacitación e instrucción. (4)

Multimedia un término que se aplica a cualquier objeto que usa simultáneamente diferentes formas de contenido informativo como texto, sonido, imágenes, animación y video para informar o entretener al

usuario. También se puede calificar como multimedia a los medios electrónicos (u otros medios) que permiten almacenar y presentar contenido multimedia. (4)

Dentro de los materiales multimedia se encuentra la multimedia interactiva, cuando el usuario tiene cierto control sobre la presentación del contenido como: qué desea ver y cuándo desea verlo. Multimedia puede dividirse en dos categorías principales: multimedia lineal y multimedia no lineal. El contenido lineal avanza sin que el usuario tenga control sobre la navegación, una película de cine sería un ejemplo de esto. El contenido no lineal le ofrece al usuario la interactividad necesaria para controlar el progreso de la presentación tal como ocurre en los videojuegos. (4)

En otras palabras, multimedia es la unión de varias medias y la interactividad entre ellas.

Aplicaciones multimedia

Las aplicaciones o programas multimedia han sido desarrollados como una interfaz amigable y sencilla de comunicación, para facilitar el acceso de todos los usuarios a las TIC. Una de las características más importantes de estos entornos es la interactividad. A diferencia de las tecnologías más clásicas (TV, radio) que permiten una interacción unidireccional, de un emisor a una masa de espectadores pasivos, el uso del ordenador interconectado mediante las redes digitales de comunicación, proporciona una comunicación bidireccional (sincrónica y asincrónica), persona-persona y persona-grupo. Se está produciendo, por tanto, un cambio hacia la comunicación entre personas y grupos que interactúan según sus intereses, conformando lo que se denomina "comunidades virtuales". El usuario de las TIC es un sujeto activo, que envía sus propios mensajes y, lo más importante, toma las decisiones sobre el proceso a seguir: secuencia, ritmo, código. (5)

Otra de las características más significativas que tienen las aplicaciones multimedia, es que posibilitan la transmisión de información por diferentes medios (texto, imagen, sonido, animaciones y videos).

Multimedia educativa

Dentro del grupo de los materiales multimedia, que integran diversos elementos textuales (secuenciales e hipertextuales) y audiovisuales (gráficos, sonidos, videos y animaciones), están los materiales multimedia educativos, que son los que se utilizan con finalidad educativa.

Recursos multimedia

Los componentes multimedia son los diversos medios de comunicación digitalizados que tiene el hombre para transmitir un mensaje con imágenes de cualquier tipo, sonidos, películas de video, y hasta el mismo texto, y constituyen un elemento crucial en la calidad de los software educativos, lo que posibilita que esta herramienta de apoyo al proceso de enseñanza-aprendizaje sea más atractiva e interactiva. (5)

Algunos sistemas trabajan con la filosofía de incrustación, que es cuando un recurso se integra físicamente en el nodo sin que exista ningún enlace para ello; la ventaja de la incrustación reside en la simplificación en la consulta y mantenimiento de recursos. Otros sistemas orientan la integración de recursos multimedia hacia el enlace: cada imagen, sonido, texto o video son considerados como nodos individuales, que posteriormente se enlazan entre sí en un único nodo "maestro" (nodos compuestos). El método anterior tiene la ventaja de que cualquier cambio en uno de los nodos repercute en los nodos compuestos. La web utiliza esta filosofía ya que para que en una página aparezca una imagen se debe especificar el fichero de la misma, así los ficheros de un servidor pueden ser considerados como nodos. (5)

La tecnología multimedia permite el acceso, a través de la computadora, a información de diversa naturaleza (una película, un sonido, una imagen y un texto), permitiendo la posibilidad de interactuar de manera dinámica con dichos recursos, superando la rigidez del sistema lineal que solo permitía transitar entre distintos menús. De este modo, habilita a aproximarse cada vez más a los modos del pensamiento humano, en virtud de la posibilidad de asociar, elegir y acceder a la información almacenada en la computadora. (5)

A partir de lo antes expuesto se puede llegar a la conclusión que los recursos multimedia en la educación son los diversos medios de comunicación digitalizados que se encuentran en los software educativos, para transmitir un mensaje o contenido. Estos recursos no son más que las imágenes, animaciones, fotografías, sonidos, fondos, videos y texto.

Hipermedia

Un sistema hipermedia es una estructura similar a la del hipertexto (organiza la información en forma de red, entendiendo esta como un modo de unión entre los elementos de información que se caracteriza

porque no existe un orden único de recorrido entre ellos, no tiene principio ni fin), con la particularidad de que esta información es multimedia (textos, imágenes, secuencias de animaciones, sonidos y video) (Anexo 1). En esta red de información se distinguen tres elementos fundamentales: los nodos, que son los contenedores de la información; las anclas, que indican las zonas de información que pueden estar relacionadas con otras; y los enlaces, que representan relaciones entre la información que contienen los nodos. (6)

En un hipermedia se pueden utilizar dos formas básicas para consultar la información que se encuentra almacenada en sus nodos: la navegación y la búsqueda. La navegación es considerada la forma primaria de acceso a la información en un hipermedia y consiste en recorrer los nodos a través de los enlaces, consultando la información que en él se almacena. En la búsqueda el usuario solicita al sistema el acceso directo a un determinado dato. (6)

La hipermedia no es más que la tecnología que permite estructurar la información de una manera no lineal, a través de nodos interconectados por enlaces. La información presentada en estos nodos podrá integrar diferentes medios tales como: texto, sonido, video y gráficos.

Hiperentorno de enseñanza – aprendizaje

Los hiperentornos de aprendizaje son concebidos para utilizarse como medios para la enseñanza y aprendizaje por profesores y alumnos. Estos hiperentornos, que tienen en común la utilización de un ambiente informático con una intención didáctica, integran los agentes humanos (alumnos, profesores) y los artificiales (recursos informáticos) y ofrecen condiciones de interacción local o a través de redes. (7)

El concepto de hiperentorno de enseñanza-aprendizaje es, como plantea Pierre Tchounikine, lo suficientemente general para incluir tanto una plataforma de educación abierta o a distancia destinada a la difusión de recursos pedagógicos como un software para el aprendizaje de una determinada competencia de álgebra. (7)

A partir de lo antes expuesto los productos de la colección La Caja Mágica se puede clasificar en un hiperentorno de enseñanza-aprendizaje, ya que fue concebido para ser utilizado por profesores y estudiantes como herramienta de apoyo en las diferentes asignaturas que se imparten en la enseñanza primaria.

Contenido

En la colección La Caja Mágica, contenido representa textos y medias (imágenes, animaciones, locuciones).

Cuestionarios interactivos

Los cuestionarios interactivos brindan un aprendizaje a partir de errores, dado que la retroalimentación inmediata que recibe el usuario le posibilita la opción de conocer el error justo en el momento en que se produce y generalmente el programa le ofrece la oportunidad de ensayar nuevas respuestas o formas de actuar para superarlos.

1.2 Análisis de soluciones similares existentes

Con el fin de facilitar el análisis y diseño de la nueva herramienta, se realiza un estudio acerca de algunas aplicaciones que son capaces de administrar y gestionar cuestionarios interactivos, juegos, actividades o ejercicios. Las analizadas son las siguientes:

Ardora

Ardora es una aplicación informática para docentes, que permite crear sus propios contenidos web en formato HTML de un modo muy sencillo. Este permite crear más de cuarenta y cinco tipos de actividades (crucigramas, sopas de letras, completar, paneles gráficos, relojes), así como más de diez tipos de páginas multimedia (galerías de imágenes o zooms de imágenes, reproductores mp3 o flv). (8)

Características:

- Capacidad de brindarles la posibilidad a los profesores de configurar las actividades como ellos estimen conveniente.
- La forma de configurar las actividades se realizan mediante formularios muy sencillos.
- Permite a los profesores centrar su esfuerzo en los elementos de la actividad no en su tratamiento informático.
- Fácil de usar.
- Rapidez a la hora de crear actividades docentes educativas.
- Permite guardar, publicar y hacer uso previo de actividades educativas.

- Permite guardar la actividad en formato HTML. (8)

Ardora no cuenta con un atractivo diseño, pero por esto no deja de ser una potente herramienta, que basa sus características en función de la creación de actividades docentes educativas.

HotPotatoes

HotPotatoes es un programa que incluye seis aplicaciones, que permite crear actividades interactivas de opción múltiple, de respuesta corta, ordenar una frase, crucigramas y juegos de ordenar y rellenar huecos para la World Wide Web. HotPotatoes es un programa gratuito, y puede ser usado para cualquier propósito o proyecto, pero no es de código abierto. (9)

Es fácilmente accesible en línea a través de Internet, gratuito para uso individual o educativo sin ánimos de lucro, siempre y cuando el material producido sea accesible públicamente. Se puede descargar de Internet, pero debe ser registrado gratuitamente para activar todas sus funcionalidades. Si el usuario lo desea puede comprar una versión para su libre distribución. (9)

Componentes:

- JBC crea ejercicios de selección múltiple.
- JCloze genera ejercicios de completar espacios en blanco.
- JCross crea crucigramas de cualquier tamaño.
- JMatch crea ejercicios de emparejamiento u ordenación.
- JMix crea ejercicios de reconstrucción de frases o párrafos a partir de palabras desordenadas.

The Master, es una herramienta que permite compilar de forma automática ejercicios de HotPotatoes en unidades didácticas. Permite enlazar todos los ejercicios de una unidad utilizando botones de navegación. (9)

CRODA

Creación De Objetos de Aprendizaje (CRODA) es una herramienta de autor desarrollada en la UCI que brinda a las comunidades de profesores la posibilidad de elaborar recursos educativos en forma de Objetos de Aprendizaje (OA), en busca de su accesibilidad, durabilidad, interoperabilidad y reusabilidad. (10)

Variadas son las estructuras que se consideran deben tener los OA, según los intereses de sus autores o de las instituciones. En CRODA se permite al docente seleccionar la estructura que desea utilizar para la elaboración de sus OA, así como la creación de una estructura propia que puede ser reutilizada por los otros profesores. (10)

Para la construcción de los OA se emplean los estándares más reconocidos internacionalmente (SCORM (del inglés *Sharable Content Object Reference Model*) y LOM (del inglés, *Learning Object Metadata*)) que contribuyen a alcanzar su interoperabilidad y gestión. Entre los elementos que componen un OA creado a partir de esta herramienta de autor se encuentran: imágenes, audios, videos, links, ecuaciones matemáticas y variados ejercicios de autoevaluación, entre otros. (10)

ExeLearning

Es una herramienta de autor gratuita de tipo código abierto, cuyo principal objetivo es facilitar la labor del docente en la creación de contenido web, sin que este posea conocimientos de programación o del lenguaje HTML. (11)

El punto más fuerte de esta herramienta de autor es posiblemente la facilidad que ofrece para la creación de material formativo con un alto grado de complejidad. Contiene diferentes recursos educativos, que permiten introducir ejercicios y lecciones con diferentes formatos como pueden ser preguntas de verdadero-falso, exámenes SCORM o artículos wiki.

Algunas características de ExeLearning son:

- Crear un sitio web con un menú lateral dinámico que asegura una navegación sencilla e intuitiva al usuario.
- Editar páginas con contenido multimedia (imágenes, vídeo, audio, animaciones, expresiones matemáticas) gracias al repertorio de herramientas de ExeLearning.
- Un repertorio de hojas de estilo.
- Exportar el proyecto como sitio web y en paquetes estándar (SCORM, IMS CP). (11)

Uno de los puntos fuertes de ExeLearning es que permite exportarlo los contenidos en archivos comprimidos SCORM, IMS o en formato web. Otra de sus ventajas es que está disponible para los sistemas operativos mayoritarios Linux, Mac y Windows.

Después de un análisis de las aplicaciones existentes, que resuelven problemas similares al presente en la colección, se puede observar, que a pesar de las funcionalidades y gran utilidad que tienen estas herramientas, no son capaces de dar solución al problema anteriormente planteado, dado que no gestionan todos los juegos, ni las tipologías que presentan dificultades con las imágenes y las retroalimentaciones en la herramienta antes desarrollada.

Ardora y HotPotato, a pesar de que gestionan tipologías de ejercicios que no son las que presentan dificultad en la colección La Caja Mágica y las de los juegos Crucigrama y Sopa de palabras, siendo esto una información que se podría utilizar como referencia para la gestión de los contenidos en la nueva aplicación; no se puede hacer un estudio de las funcionalidades por ambas ser aplicaciones que no son de código abierto, es decir, no se puede estudiar, modificar, mejorar ni copiar el código que generan. ExeLearning y CRODA no posibilitan el almacenamiento de los contenidos en BD de modo que sean útiles a la colección La Caja Mágica, siendo necesario guardar la información de los juegos y ejercicios de esta, sino que los exporta en paquetes SCORM, los cuales no son interpretados por la colección. Por tanto, lo que se necesita para solucionar el problema planteado es una aplicación capaz de gestionar los contenidos que no se incluyen en el panel de administración de contenidos de la colección La Caja Mágica.

1.3 Tendencias y tecnologías actuales

La herramienta que se diseñó es la versión 1.1 de GCon y tributa al proyecto productivo Multisaber-Navegante, por lo que las herramientas y tecnologías a utilizar están establecidas en las políticas del proyecto. En este epígrafe se expone un resumen y las características más significativas de las herramientas y tecnologías que serán las utilizadas para dar solución al problema a resolver.

1.4.1 Metodología de desarrollo

Para obtener un correcto desarrollo del software, existen metodologías, que son un conjunto de procedimientos, técnicas y pasos a seguir, que ayudan a la documentación para el desarrollo de este. Establecen un conjunto de actividades a realizar que definen cómo se debe hacer el software, quién debe hacer cada actividad, cuándo hacerla y qué se debe hacer. Estas metodologías permiten la obtención de un software de calidad y la construcción del producto en el tiempo planificado.

Existen varias metodologías orientadas al proceso de desarrollo de software, a continuación se mencionan algunas de las más significativas: Programación Extrema (XP, por sus siglas en inglés, eXtreme Programming), SCRUM y Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés, Rational Unified Process).

RUP

Es un proceso de ingeniería del software. Proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo de software. Esta metodología tiene como propósito el aseguramiento de la producción de software de alta calidad.

Se divide en cuatro fases del desarrollo del software, estas son:

- Inicio, determinar la visión del proyecto.
- Elaboración, determinar la arquitectura óptima.
- Construcción, obtener la capacidad operacional inicial.
- Transición, llegar a obtener el *release* (liberación) del proyecto.

RUP es un proceso que tiene tres características esenciales, está dirigido por los casos de uso, está centrado en la arquitectura, y es iterativo e incremental que define quién debe hacer qué, cómo y cuándo. Está fundamentada en un enfoque orientado a modelos de desarrollo basado en componentes. (12)

En su modelación se definen como elementos principales los siguientes:

- Trabajadores (**quién**): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Son los que realizan las actividades.
- Actividades (**cómo**): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- Artefactos (**qué**): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- Flujo de actividades (**cuándo**): Secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable. (12)

Define 9 disciplinas a realizar en cada fase del proyecto.

Disciplinas de ingeniería:

- **Modelo del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será construido a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación o despliegue:** Produce con éxito distribuciones del producto y las entrega a los usuarios finales. (12)

Disciplinas de soporte:

- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. (12) (Anexo 2)

RUP tiene bien definido y organizado el proceso de desarrollo de software, lo que permite que los involucrados estén claro de los pasos a seguir para obtener un producto con calidad. (12)

Para lograr un correcto desarrollo del software se estableció en las políticas del proyecto Multisaber-Navegante la utilización de la metodología RUP, puesto que captura las mejores prácticas del conocimiento de líderes en ingeniería de software y facilita a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad. Es orientada a objetos y genera una

abundante documentación que ayuda a entender y mantener el sistema.

1.4.2 Lenguaje de modelado: Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Se define como un “Lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software”. (13)

Incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. La utilización de UML es independiente del lenguaje de programación y de las características de los proyectos, pues ha sido diseñado para modelar cualquier tipo de proyecto. (13)

El lenguaje de modelado establecido en las políticas del proyecto Multisaber-Navegante es UML, por las características antes descritas, además que es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, de un modo estándar incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema.

1.4.3 Herramienta CASE: Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Puede ser usada en varios idiomas. Además es multiplataforma, es decir, que permite ser utilizada en cualquier sistema operativo. (14)

Características de Visual Paradigm:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor

calidad.

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales Entornos de Desarrollo Integrado (IDEs).
- Disponibilidad en múltiples plataformas. (14)

Por las razones antes expuestas, que hacen que esta herramienta sea potente y utilizada a nivel mundial, debido a la cantidad de funcionalidades y ventajas sobre otras de su tipo; es seleccionada por el proyecto Multisaber-Navegante para llevar a cabo el modelado visual del sistema.

1.4.4 Lenguajes del lado del cliente: JavaScript

Este es un lenguaje interpretado, no requiere compilación. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript. Los códigos JavaScript se insertan directamente en el mismo documento HTML. (15) Es ideal para agregar funciones rápidas a una página web y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador.

Ventajas:

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.

Desventajas:

- Código visible por cualquier usuario.
- El código debe descargarse completamente.

- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado vulnerabilidad (XSS, por sus siglas en inglés, Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS). (15)

1.4.5 Lenguajes del lado del cliente: HTML

Desde el surgimiento de Internet se han publicado sitios web gracias al Lenguaje de Marcado de Hipertexto (HTML, por sus siglas en inglés, HyperText Markup Language). Es un lenguaje estático para el desarrollo de sitios web. Nos permite definir de forma estructurada las páginas Web, así como todo su contenido: texto, imágenes, sonido, entre otros. Los archivos pueden tener las extensiones (html, htm). (15)

Ventajas:

- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los navegadores.

Desventajas:

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas. (15)

1.4.6 Lenguajes del lado del cliente: CSS

Hojas de Estilo en Cascada (CSS, por sus siglas en inglés, Cascading Style Sheets), es un lenguaje de hojas de estilos elaborado para controlar la fachada o presentación de los documentos electrónicos definidos con HTML y XHTML, es decir, describe cómo se va a mostrar un documento en pantalla.

Permite separar la presentación de sus contenidos en los documentos escritos en HTML, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra, modificando la apariencia de

una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.

1.4.7 Lenguajes del lado del servidor: PHP

Es un lenguaje interpretado de alto nivel que se utiliza para la creación de sitios web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). (15)

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. Para su funcionamiento necesita tener instalado servidores web como Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. (15)

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente, por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.

- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación. (15)

Los lenguajes establecidos en las políticas del proyecto Multisaber-Navegante son: del lado del cliente: HTML que maqueta y diseña los elementos visuales del programa, JavaScript que agrega las funcionalidades en las interfaces de usuario, permitiendo hacer validaciones previas y del lado del servidor es PHP ya que este se encarga de realizar las acciones del lado del servidor asociadas a la lógica del negocio, es multiplataforma, tiene capacidad de conexión con la mayoría de los manejadores de la BD y es libre. El seleccionado para darle estilo a los elementos visuales del programa es CSS, ya que se encargará de separar la presentación del contenido.

1.4.8 Arquitectura y patrones

Las técnicas metodológicas desarrolladas con el fin de facilitar la programación se engloban dentro de la llamada Arquitectura de Software o Arquitectura lógica y su definición oficial en la IEEE Std 1471-2000 que reza así: “La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. Los patrones de arquitectura son un modelo que proporcionan reglas y guías para organizar un sistema de software. (16)

Arquitectura tres Capas

Este tipo de arquitectura se utiliza para diseñar en tres capas, con el objetivo de no colocar el código en las interfaces del usuario del sistema. Para dar solución a este inconveniente, la idea es tener tres niveles de funcionalidad bien determinados, teniendo como objetivo primordial la separación de la capa de presentación, capa de negocio y la de datos. (Anexo 3)

Capa de Presentación (o interfaz del usuario): Es lo que ve el usuario (también se le denomina "capa de usuario"), y con la que interactúa directamente. Esta capa captura la información entrada por el usuario y hace las peticiones a la capa de negocio. Luego muestra al usuario la respuesta. También se le conoce como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Se comunica directamente con la capa de negocio.

Capa Lógica de Funcionalidad/Negocio: Es donde residen los programas que se ejecutan, se reciben

las peticiones del usuario y se envían las respuestas. Se denomina capa de negocio (e incluso de lógica del negocio) o de aplicación porque es aquí donde se establecen todas las reglas que deben cumplirse, es decir, contiene la lógica del negocio. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de Datos: Esta capa incluye la BD que contiene todos los datos. Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

La arquitectura definida en las políticas del proyecto Multisaber-Navegante para la aplicación fue una arquitectura tres capas ya que permite mostrar distintos tipos de vistas sin afectar al modelo. Esta arquitectura se diferencia de la tradicional cliente/servidor, dado que la aplicación reside en la capa media (servidor de aplicación).

1.4.9 Entorno de desarrollo integrado

Un IDE es un programa compuesto por una serie de herramientas a utilizar por los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. (17)

Las herramientas que normalmente componen un IDE son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones. (17)

Hoy en día los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python y Visual Basic, entre otros). Además es posible que un mismo entorno de desarrollo tenga la posibilidad de utilizar varios lenguajes de programación, como es el caso de Eclipse. (17)

NetBeans IDE

El IDE NetBeans es un IDE de código abierto y gratuito, sin restricciones de utilización. Proporciona a los desarrolladores múltiples herramientas que permiten la creación de aplicaciones empresariales, tanto para la Web como para aplicaciones móviles. Está desarrollado en Java pero puede servir para cualquier otro lenguaje. El IDE se puede complementar con varios paquetes que le proporcionan funciones adicionales, algunos de estos paquetes añaden al NetBeans las herramientas necesarias para escribir, depurar y probar código. NetBeans brinda un excelente soporte para PHP y JavaScript que incluye resaltado de sintaxis, autocompletado, ayuda, control de versiones, gestión de ficheros, pre-visualización, refactorizado, debugger. (18)

El NetBeans es un producto de código abierto, que dispone de forma gratuita todos los beneficios del software, brinda soporte a JavaScript y completamiento de código HTML y CSS. Con NetBeans IDE se pueden desarrollar aplicaciones de escritorio, para la web y para teléfonos celulares, utilizando lenguajes de programación como Java, C/C++, Ruby on Rails, PHP, Python y Javascript. Por lo antes expuesto hace que sea el IDE seleccionado en las políticas del proyecto Multisaber-Navegante.

1.4.10 Sistemas gestores de base de datos

En la actualidad, cada vez existe una mayor demanda de datos, proviniendo principalmente de las grandes empresas. Cada día surgen nuevos cambios que hacen que el avance de las TIC sea acelerado, con lo cual estas demandas se disparan debido al acceso multitudinario a Internet. Por ello, las bases de datos se reconocen como una de las principales aplicaciones de la informática, en las cuales se conoce como dato cualquier elemento informativo que tenga relevancia para un usuario.

Un Sistema Gestor de Base de Datos (SGBD) o DBMA (Data Base Management System) es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Un SGBD permite definir una base de datos (especificar tipos, estructuras y restricciones de datos), construirla (guardar los datos en algún medio controlado por el mismo) y manipularla (realizar consultas, actualizarla y generar informes). Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server. (19)

Las características de un SGBD son:

- **Abstracción de la información.** Los detalles de almacenamientos son transparentes al usuario.
Redundancia mínima. Evitar la aparición de información repetida o redundante.
- **Consistencia.** Cuando no se ha logrado esta redundancia nula, es necesario tener en cuenta que aquella información que aparece repetida se actualice de forma coherente.
- **Seguridad.** La información almacenada puede tener un gran valor, por lo que se debe garantizar que esta información se encuentra segura frente a usuarios con otras intenciones. Habitualmente, poseen un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Integridad.** Proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra ocurrencia capaz de dañar la información almacenada.
- **Respaldo y recuperación.** Deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- **Control de la concurrencia.** Controlar los accesos concurrentes a la información para evitar las inconsistencias. (19)

MySQL

Es un sistema de gestión de bases de datos relacional, licenciado bajo la Licencia Pública General de GNU, creada por la empresa sueca MySQL AB. Las principales características de este SGBD son las siguientes:

- Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- Capacidad de gestionar y almacenar grandes cantidades de datos.
- Rápido y fácil de usar.
- Gran portabilidad entre sistemas.
- Fácil de instalar y configurar.
- Gestión de usuarios y contraseñas, manteniendo un buen nivel de seguridad en los datos.
- Cuenta con una infinidad de librerías y herramientas que facilitan su uso. (20)

1.4.11 Servidores web

Un servidor web es un programa o software que atiende las peticiones de los usuarios o navegadores. El

término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos. (23) Entre los servidores web más utilizados se encuentran el Internet Information Server (IIS) y el Apache.

Apache

Este servidor es uno de los mayores triunfos del software libre, que tanto gusta a los usuarios de Linux. Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos. Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para solución de los mismos. (22)

1.5 Conclusiones

En este capítulo quedó plasmado todo lo correspondiente a la base teórica de la investigación, donde se consiguió realizar una valoración del estado del arte, mediante el análisis de las soluciones similares. Para lograr un correcto desarrollo del software quedó establecido en las políticas del proyecto Multisaber-Navegante la utilización de las siguientes herramientas y tecnologías: la metodología RUP, ya que implementa las mejores prácticas de Ingeniería de Software para poder lograr el desarrollo del flujo de trabajo de Análisis y Diseño con éxito; HTML y JavaScript como lenguajes del lado del cliente y del lado del servidor PHP; Visual Paradigm como herramienta CASE que dará soporte al modelado visual con UML; como entorno de desarrollo el NetBeans; el patrón de arquitectura definido es tres capas; como SGBD MySQL y el servidor web Apache.

Capítulo 2. Descripción de la solución propuesta

2.1 Introducción

En el presente capítulo se realiza una descripción del entorno donde se usará la aplicación. Se aborda el funcionamiento general del subsistema mediante un modelo de dominio, retocando cada uno de los conceptos que lo conforman para que sea más entendible el contexto en que se ubica el sistema. Se especifican los requisitos funcionales y no funcionales del sistema propuesto, se identifican los actores, los casos de uso y las relaciones existentes entre ellos a través del modelo de casos de uso del sistema.

2.2 Modelo de Dominio

Debido a que los procesos del negocio no poseen fronteras bien establecidas donde se logren ver claramente, cuáles son los procesos, personas y actividades involucradas, se decide realizar un modelo de dominio o modelo conceptual, en el cual se capturan los tipos de objetos o clases más importantes que existen, o los eventos que suceden en el entorno donde estará el sistema.

Un modelo conceptual explica los conceptos más significativos en el dominio del problema, realizado con las reglas de UML se presenta como un diagrama de clases que contiene, no conceptos propios de un sistema de software, sino de la propia realidad física. Este modelo ayuda a comprender los conceptos que utilizan los usuarios.

El modelo de dominio que a continuación se presenta constituye una representación gráfica de los conceptos más importantes que forman parte de la versión 1.1 de la herramienta GCon para gestionar los contenidos de la colección La Caja Mágica.

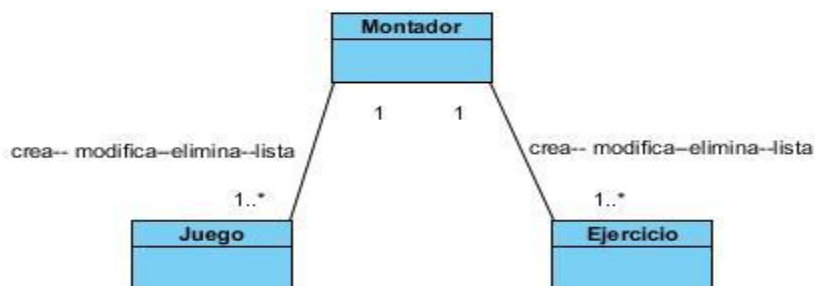


Ilustración 1. Modelo del dominio.

El siguiente diccionario o glosario de datos está formado por los conceptos identificados en el contexto del problema y se utilizan en la confección del modelo de dominio. Tiene como objetivo principal permitir la comprensión del entorno y del problema.

Ejercicio: actividad interactiva destinada a que los niños adquieran, desarrollen o apliquen conocimientos sobre los temas que se abordan en la colección La Caja Mágica.

Juego: es una actividad recreativa que cuenta con la participación de uno hasta 4 jugadores. Su función principal es proporcionar entretenimiento y diversión, combinados con contenidos educativos de la colección La Caja Mágica.

Montador: persona que interactúa con el sistema para crear, modificar, eliminar o listar el contenido de la colección La Caja Mágica.

2.3 Especificación de requerimientos

Los requerimientos son las características y cualidades que el sistema debe tener y se agrupan en funcionales y no funcionales. Los requerimientos funcionales definen las acciones que el sistema será capaz de realizar, son capacidades o condiciones que debe cumplir. Los requerimientos no funcionales describen las características del sistema como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad y estándares. Estos requerimientos hacen que el producto tenga las características que quiere el cliente, que sea atractivo, rápido, confiable y usable. Una vez definidos e identificados los mismos, se puede pasar a modelar el sistema. (12)

2.3.1 Requerimientos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir para darle solución al problema identificado. A continuación se muestran los requerimientos funcionales identificados en el levantamiento de requisitos:

RF1 Mostrar cantidad de ejercicios.

El sistema debe mostrar una pantalla donde se encuentran varias tablas, donde el contenido en dichas tablas está agrupado por tipología, nivel y tema.

RF2 Crear ejercicio de tipología Selección simple.

El sistema debe mostrar la pantalla para crear un ejercicio de Selección simple, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF3 Crear ejercicio de tipología Selección múltiple.

El sistema debe mostrar la pantalla para crear un ejercicio de Selección múltiple, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

FR4 Crear ejercicio de tipología Verdadero o falso.

El sistema debe mostrar la pantalla para crear un ejercicio de Verdadero o falso, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF5 Crear ejercicio de tipología Completar por desplazamiento.

El sistema debe mostrar la pantalla para crear un ejercicio de Completar por desplazamiento, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF6 Crear ejercicio de tipología Completar por escritura.

El sistema debe mostrar la pantalla para crear un ejercicio de Completar por escritura, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF7 Crear ejercicio de tipología Seleccionar palabra.

El sistema debe mostrar la pantalla para crear un ejercicio de Seleccionar palabra, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF8 Crear ejercicio de tipología Ordenar secuencia de imágenes.

El sistema debe mostrar la pantalla para crear un ejercicio de Ordenar secuencia de imágenes,

permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF9 Crear ejercicio de tipología Enlazar columnas.

El sistema debe mostrar la pantalla para crear un ejercicio de Enlazar columnas, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF10 Crear ejercicio de tipología Formar conjuntos.

El sistema debe mostrar la pantalla para crear un ejercicio de Formar conjuntos, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF11 Crear ejercicio de tipología Ordenar oraciones.

El sistema debe mostrar la pantalla para crear un ejercicio de Ordenar oraciones, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF12 Crear ejercicio de tipología Seleccionar fragmento de texto.

El sistema debe mostrar la pantalla para crear un ejercicio de Seleccionar fragmento de texto, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF13 Crear ejercicio de tipología Identificar objeto.

El sistema debe mostrar la pantalla para crear un ejercicio de Identificar objeto, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF14 Crear ejercicio de tipología Identificar clase de objeto.

El sistema debe mostrar la pantalla para crear un ejercicio de Identificar clases de objeto, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF15 Crear ejercicio de tipología Ordenar pasos.

El sistema debe mostrar la pantalla para crear un ejercicio de Ordenar pasos, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF16 Crear ejercicio de tipología Dividir palabras.

El sistema debe mostrar la pantalla para crear un ejercicio de Dividir palabras, permitiendo introducir los datos del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF17 Modificar ejercicio.

El sistema debe mostrar la pantalla para modificar el ejercicio seleccionado, permitiendo modificar los datos del ejercicio y guardar, una vez que el montador acceda a esa opción.

RF18 Eliminar ejercicio.

El sistema debe mostrar la pantalla para eliminar el ejercicio seleccionado, una vez que el montador acceda a esa opción.

RF19 Especificar datos generales de ejercicio.

El sistema debe mostrar la pantalla donde están incluidos todos los datos generales para crear un ejercicio, permitiendo introducir los datos generales del ejercicio y creando el ejercicio a partir del registro del mismo, una vez que el montador acceda a crear uno.

RF20 Mostrar listado de ejercicios.

El sistema debe mostrar la pantalla con el listado de los ejercicios según su tipología, nivel y tema, una vez que el montador acceda a esa opción.

RF21 Mostrar ayuda.

El sistema debe mostrar la pantalla de la ayuda, una vez que el montador acceda a esta.

RF22 Eliminar selección de ejercicios.

El sistema debe permitir que el montador seleccione los ejercicios que desea eliminar y eliminarlos.

RF23 Guardar ejercicio.

El sistema debe permitir al montador la opción de guardar el ejercicio.

RF24 Crear juego de Crucigrama.

El sistema debe mostrar la pantalla para crear un juego de Crucigrama, permitiendo introducir los datos del juego y creando el juego a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF25 Crear juego de Adivinanzas.

El sistema debe mostrar la pantalla para crear un ejercicio de Adivinanzas, permitiendo introducir los datos del juego y creando el juego a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF26 Crear juego de Sopa de palabras.

El sistema debe mostrar la pantalla para crear un ejercicio de Sopa de palabras, permitiendo introducir los datos del juego y creando el juego a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF27 Crear juego de Descubre la imagen 2.

El sistema debe mostrar la pantalla para crear un ejercicio de Descubre la imagen 2, permitiendo introducir los datos del juego y creando el juego a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF28 Crear juego de Escribiendo con el teclado.

El sistema debe mostrar la pantalla para crear un ejercicio de Escribiendo con el teclado, permitiendo introducir los datos del juego y creando el juego a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF29 Crear juego de Escalera de palabras.

El sistema debe mostrar la pantalla para crear un ejercicio de Escalera de palabras, permitiendo introducir los datos del juego y creando el juego a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF30 Guardar juego.

El sistema debe permitir al montador la opción de guardar el juego.

RF31 Modificar juego.

El sistema debe mostrar la pantalla para modificar el juego seleccionado, permitiendo modificar los datos del juego y guardar, una vez que el montador acceda a esa opción.

RF32 Eliminar juego.

El sistema debe mostrar la pantalla para eliminar el juego seleccionado, una vez que el montador acceda a esa opción.

RF33 Mostrar listado de contenidos de juego.

El sistema debe mostrar la pantalla con el listado de los juegos, una vez que el montador acceda a esa opción.

RF34 Eliminar selección de contenidos de juego.

El sistema debe permitir que el montador seleccione los juegos o los temas que desea eliminar y eliminarlos.

RF35 Crear tema.

El sistema debe mostrar la pantalla para crear un tema, permitiendo introducir los datos del tema y creando el tema a partir del registro del mismo, una vez que el montador acceda a esa opción.

RF36 Guardar tema.

El sistema debe permitir al montador la opción de guardar el tema.

RF37 Modificar tema.

El sistema debe mostrar la pantalla para modificar el tema seleccionado, permitiendo modificar los datos del tema y guardar, una vez que el montador acceda a esa opción.

RF38 Eliminar tema.

El sistema debe mostrar la pantalla para eliminar el tema seleccionado, una vez que el montador acceda a esa opción.

2.3.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que la herramienta debe tener, las

características que hacen que la herramienta sea atractiva, usable, rápido o confiable. Estos son fundamentales para el éxito de una herramienta y se vinculan a los requerimientos funcionales, es decir, una vez definido que debe hacer la herramienta se puede determinar cómo ha de comportarse o qué cualidades debe tener.

1. Requisitos de hardware

Se necesita como mínimo un procesador PENTIUM 233 MHz. La resolución de pantalla debe ser de 800x600 y 24 bits en lo adelante. Memoria de 64 MB RAM (Recomendado: 128 MB RAM o mejor). Disco duro de 1GB y dispositivo de red de al menos 10 Mbits.

2. Requisitos de software

Se necesitará para lograr ejecutar la aplicación una computadora personal con el navegador instalado Mozilla Firefox 3.6 en adelante.

3. Requisitos de restricciones en el diseño y la implementación

Los lenguajes a emplear son JavaScript, HTML y PHP. La herramienta de desarrollo de la aplicación será el NetBeans, como servidor Apache 2.x y como SGBD MySQL 5.0.

4. Requisitos de apariencia e interfaz externa

El diseño de las interfaces debe ser amigable y sencillo. Se utilizará un vocabulario reconocido por los usuarios y el sistema proporcionará claridad y una adecuada organización en la información, garantizando una buena comprensión.

5. Requerimientos de usabilidad

Los usuarios que utilizarán el sistema deberán tener al menos un conocimiento básico de cómo se maneja una computadora y de cómo se trabaja con la aplicación. Además cada ejercicio y juego cuenta con una ayuda que se ubicará como un servicio en el mismo y será escrita en lenguaje español lo más sencilla y asequible posible para facilitar su entendimiento por parte de los usuarios.

6. Requisitos legales

Las imágenes que se utilicen en el producto deberán tener el permiso legal de sus autores y la aprobación para usar las en el software.

7. Requisitos de soporte

La tarjeta de video donde se ejecutará el software debe ser integrada de 128 Mb (como mínimo).

8. Requisitos de portabilidad

El software podrá ser ejecutado bajo cualquier Sistema Operativo.

2.4 Modelo de casos de uso del sistema

2.4.1 Actores del sistema

Tabla 1. Actor del sistema

Actor	Justificación
Montador	Es la persona que va a interactuar con la aplicación para insertar, modificar, eliminar o ver el contenido que contiene la aplicación.

2.4.2 Diagrama de Casos de Uso del Sistema

Se divide el diagrama de casos de uso en 4 diagramas, para una mejor visualización y comprensión en el documento.

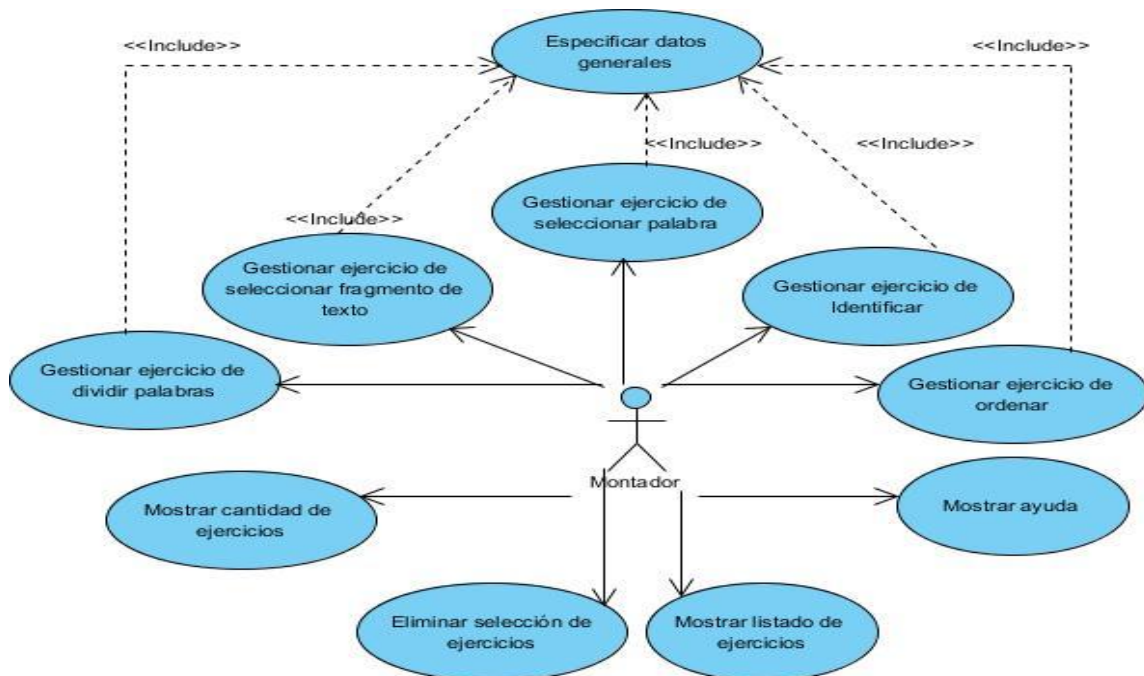


Ilustración 2. Diagrama 1 de casos de uso del sistema.

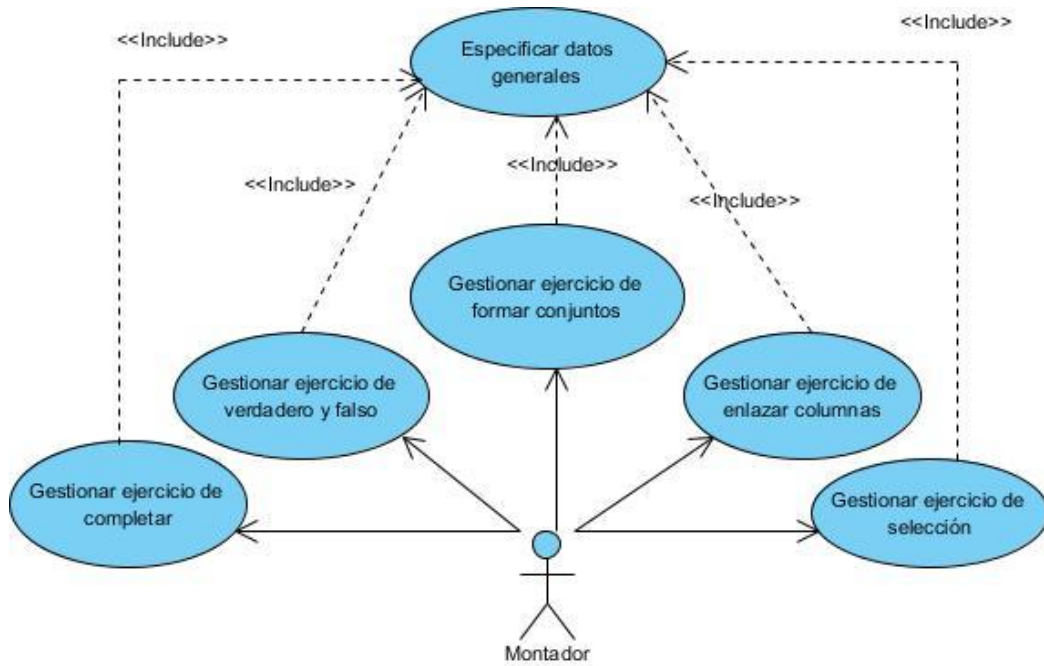


Ilustración 3. Diagrama 2 de casos de uso del sistema.

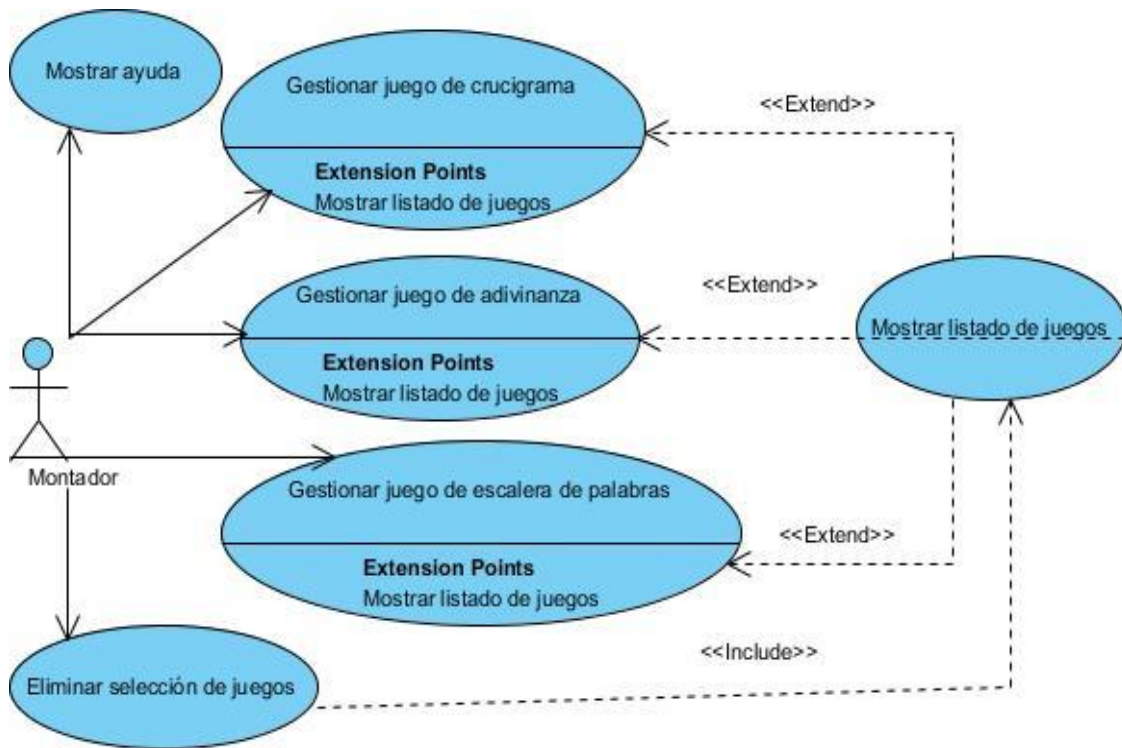


Ilustración 4. Diagrama 3 de casos de uso del sistema.

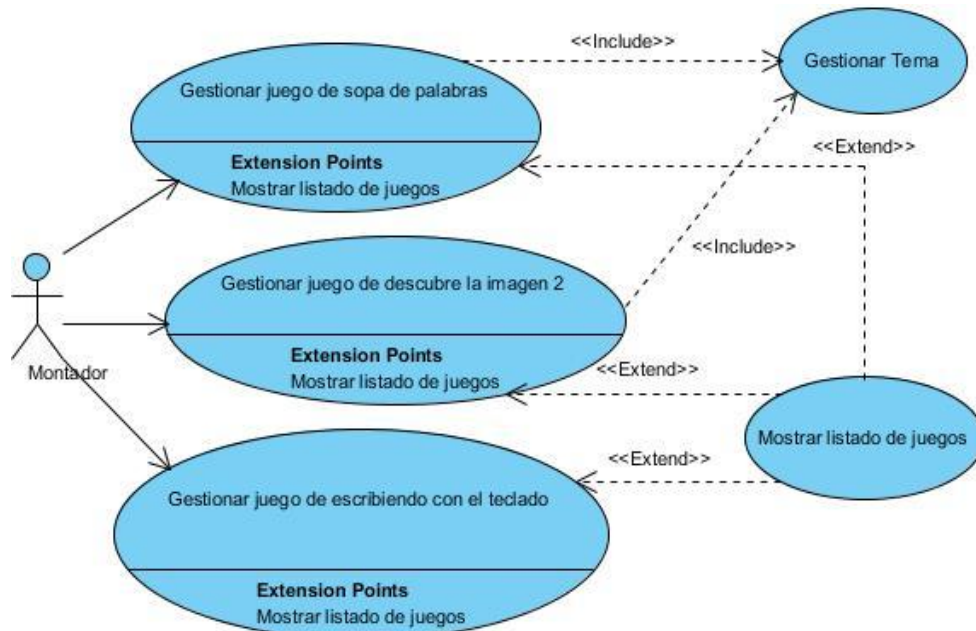


Ilustración 5. Diagrama 4 de casos de uso del sistema.

2.4.3 Descripción de casos de uso del sistema

A continuación se realiza un resumen de las descripciones de los casos de uso del sistema y se muestra el prototipo de interfaz de usuario asociado, los restantes resúmenes se encuentran en el Anexo 4. Todas las descripciones íntegras se encuentran en el Expediente de Proyecto en la plantilla Modelo del Sistema como apéndice de la investigación.

Tabla 2. Gestionar juego de Crucigrama.

Caso de Uso:	Gestionar juego de Crucigrama.
Actores:	Montador
Resumen:	El caso de uso se inicia cuando el actor accede al sistema con el objetivo de Crear, Modificar o Eliminar un juego de Crucigrama. El sistema permite seleccionar la acción que desea hacer sobre el mismo. El actor selecciona Crear un juego de Crucigrama, el sistema permite introducir los datos correspondientes y lo crea. En caso que el actor selecciona Modificar, el sistema muestra los datos de modo editable permitiendo modificar y actualizar los valores. En caso que el actor acceda a la opción Eliminar, el sistema elimina el juego. Terminando así el caso de uso.

Precondiciones:	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Referencias	RF21, RF24, RF30, RF31, RF32, RF33, RF34
Prioridad	Crítico



Ilustración 6. Prototipo de interfaz gráfica de usuario: Juego de Crucigrama.

Tabla 3. Gestionar juego de Adivinanza.

Caso de Uso:	Gestionar juego de Adivinanza.
Actores:	Montador
Resumen:	El caso de uso se inicia cuando el actor accede al sistema con el objetivo de Crear, Modificar o Eliminar un juego de Adivinanza. El sistema permite seleccionar la acción que desea hacer sobre el mismo. El actor selecciona Crear un juego de Adivinanza, el sistema permite introducir los datos correspondientes y lo crea. En caso que el actor selecciona Modificar, el sistema muestra los datos de modo editable permitiendo modificar y actualizar los valores. En caso que el actor acceda a la opción Eliminar, el sistema elimina el juego. Terminando así el caso de uso.
Precondiciones:	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Referencias	RF21, RF25, RF30, RF31, RF32, RF33, RF34
Prioridad	Crítico



Ilustración 7. Prototipo de interfaz gráfica de usuario: Juego de Adivinanza.

Tabla 4. Gestionar juego de Sopa de palabras.

Caso de Uso:	Gestionar juego de Sopa de palabras.
Actores:	Montador
Resumen:	El caso de uso se inicia cuando el actor accede al sistema con el objetivo de Crear, Modificar o Eliminar un juego de Sopa de palabras. El sistema permite seleccionar la acción que desea hacer sobre el mismo. El actor selecciona Crear un juego de Sopa de palabras, el sistema permite introducir los datos correspondientes y lo crea. En caso que el actor selecciona Modificar, el sistema muestra los datos de modo editable permitiendo modificar y actualizar los valores. En caso que el actor acceda a la opción Eliminar, el sistema elimina el juego. Terminando así el caso de uso.
Precondiciones:	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Referencias	RF21, RF26, RF30, RF31, RF32, RF33, RF34, RF35
Prioridad	Crítico

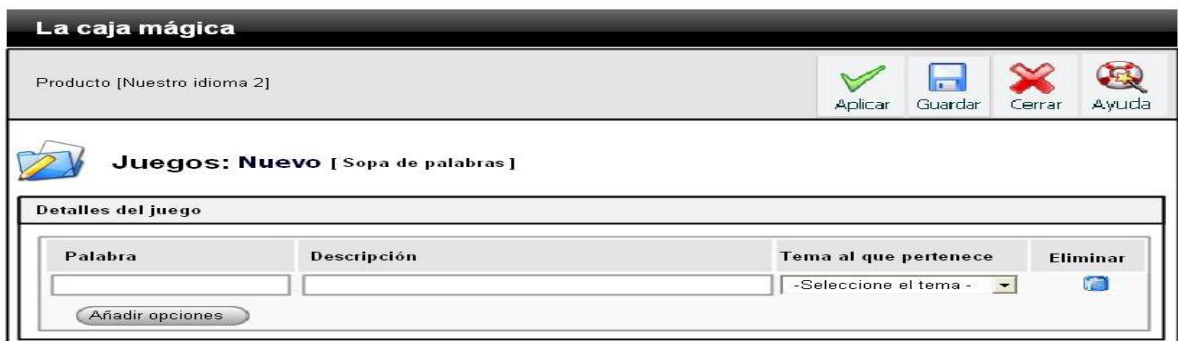


Ilustración 8. Prototipo de interfaz gráfica de usuario: Juego de Sopa de palabras.

Tabla 5. Gestionar juego de Descubre la imagen 2.

Caso de Uso:	Gestionar juego de Descubre la imagen 2.
Actores:	Montador
Resumen:	El caso de uso se inicia cuando el actor accede al sistema con el objetivo de Crear, Modificar o Eliminar un juego de Descubre la imagen 2. El sistema permite seleccionar la acción que desea hacer sobre el mismo. El actor selecciona Crear un juego de Descubre la imagen 2, el sistema permite introducir los datos correspondientes y lo crea. En caso que el actor selecciona Modificar, el sistema muestra los datos de modo editable permitiendo modificar y actualizar los valores. En caso que el actor acceda a la opción Eliminar, el sistema elimina el juego. Terminando así el caso de uso.
Precondiciones:	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Referencias	RF21, RF27, RF30, RF31, RF32, RF33, RF34, RF35
Prioridad	Crítico



Ilustración 9. Prototipo de interfaz gráfica de usuario: Juego de Descubre la imagen 2.

Tabla 6. Gestionar juego de Escribiendo con el teclado.

Caso de Uso:	Gestionar juego de Escribiendo con el teclado.
Actores:	Montador
Resumen:	El caso de uso se inicia cuando el actor accede al sistema con el objetivo de Crear, Modificar o Eliminar un juego de Escribiendo con el teclado. El sistema permite seleccionar la acción que desea hacer sobre el mismo. El actor selecciona Crear un juego de Escribiendo con el teclado, el sistema permite introducir los datos correspondientes y lo crea. En caso que el actor selecciona Modificar, el sistema muestra los datos de modo editable permitiendo

	modificar y actualizar los valores. En caso que el actor acceda a la opción Eliminar, el sistema elimina el juego. Terminando así el caso de uso.
Precondiciones:	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Referencias	RF21, RF28, RF30, RF31, RF32, RF33, RF34
Prioridad	Crítico



Ilustración 10. Prototipo de interfaz gráfica de usuario: Juego de Escribiendo con el teclado.

Tabla 7. Gestionar juego de Escalera de palabras.

Caso de Uso:	Gestionar juego de Escalera de palabras.
Actores:	Montador
Resumen:	El caso de uso se inicia cuando el actor accede al sistema con el objetivo de Crear, Modificar o Eliminar un juego de Escalera de palabras. El sistema permite seleccionar la acción que desea hacer sobre el mismo. El actor selecciona Crear un juego de Escalera de palabras, el sistema permite introducir los datos correspondientes y lo crea. En caso que el actor selecciona Modificar, el sistema muestra los datos de modo editable permitiendo modificar y actualizar los valores. En caso que el actor acceda a la opción Eliminar, el sistema elimina el juego. Terminando así el caso de uso.
Precondiciones:	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Referencias	RF21, RF28, RF30, RF31, RF32, RF33, RF34
Prioridad	Crítico

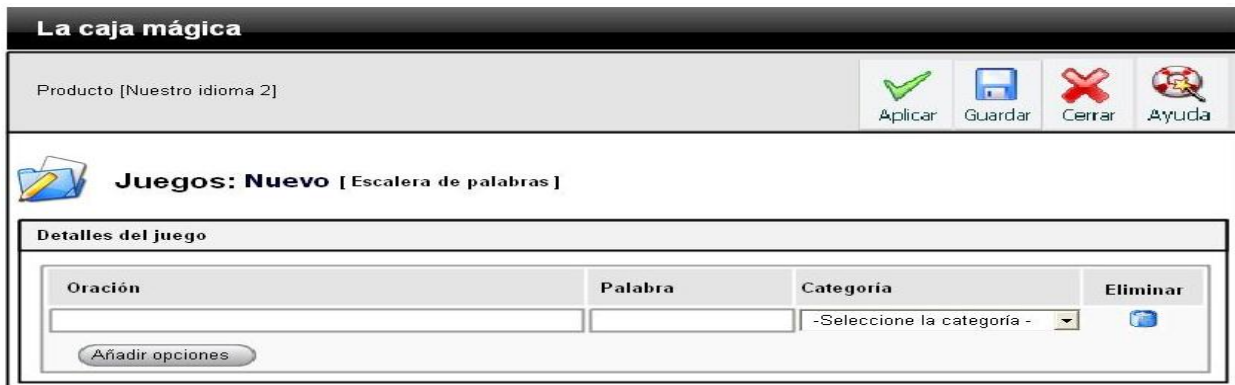


Ilustración 11. Prototipo de interfaz gráfica de usuario: Juego de Escalera de palabras.

Tabla 8. Gestionar tema.

Caso de Uso:	Gestionar Tema.
Actores:	Montador
Resumen:	El caso de uso se inicia cuando el actor accede al sistema con el objetivo de Crear, Modificar o Eliminar un tema. El sistema permite seleccionar la acción que desea hacer sobre el mismo. El actor selecciona Crear un tema, el sistema permite introducir los datos correspondientes y lo crea. En caso que el actor selecciona Modificar, el sistema muestra los datos de modo editable permitiendo modificar y actualizar los valores. En caso que el actor acceda a la opción Eliminar, el sistema elimina el tema. Terminando así el caso de uso.
Precondiciones:	Debe haberse generado el escritorio de trabajo del usuario autenticado.
Referencias	RF35
Prioridad	Crítico

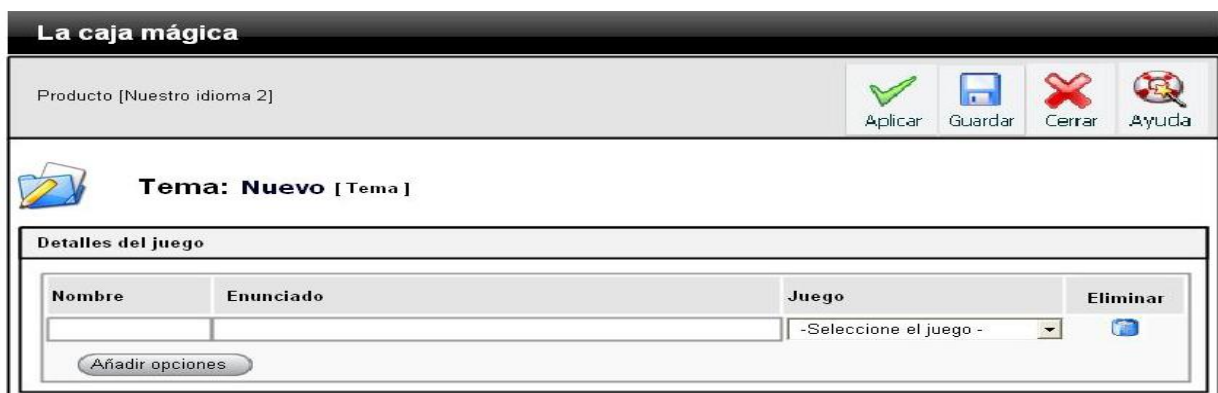


Ilustración 12. Prototipo de interfaz gráfica de usuario: Gestionar tema.

2.5 Conclusiones

En este capítulo se fundamentaron las descripciones y especificaciones del dominio del negocio y del sistema. Se identificaron los requerimientos funcionales y no funcionales, agrupando los requerimientos funcionales en casos de uso obteniendo así el diagrama de casos de uso del sistema y las descripciones de los diferentes casos de uso.

Capítulo 3: Análisis y Diseño del Sistema

3.1 Introducción

En el presente capítulo se realizó el Análisis y Diseño del sistema propuesto. Del diagrama de casos de uso, obtenido como parte del flujo de trabajo requerimientos, se sigue con el flujo que continúa el ciclo de vida de RUP: Análisis y Diseño. De esta forma se modelan los artefactos necesarios, los cuales son modelo de análisis, modelo de diseño y modelo de datos.

3.2 Modelo de Análisis

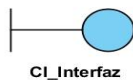
En el análisis, es importante realizar un estudio de los requisitos obtenidos, con el objetivo de tener una mejor comprensión antes de diseñar el software, garantizando así una arquitectura robusta, eficaz, eficiente y capaz de sobrevivir a cambios.

El modelo de análisis es utilizado fundamentalmente por los desarrolladores para percibir, cómo darle forma al sistema, es decir cómo debería ser diseñado e implementado, proporcionando una estructura centrada en el mantenimiento de aspectos tales como la flexibilidad ante los cambios y la reutilización. Esta estructura creada es la que hace posible tener una correcta entrada en las actividades de diseño e implementación. (12)

3.2.1 Diagrama de Clases del Análisis

En el modelo de análisis están contenidas las clases de análisis, que no es más que una vista estática de las clases y las asociaciones entre las mismas. En la construcción del diagrama de clases del análisis se utilizan tres tipos de clases y los siguientes estereotipos son los que las identifican:

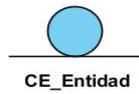
Estereotipo



Modelan la interacción entre el sistema y sus actores.



Coordinan la realización de los casos de uso.



Modelan información que posee larga vida y que es a menudo persistente.

A continuación se muestran los diagramas de clases de análisis correspondientes a los casos de uso descritos anteriormente. Los restantes diagramas se muestran en el Anexo 5.

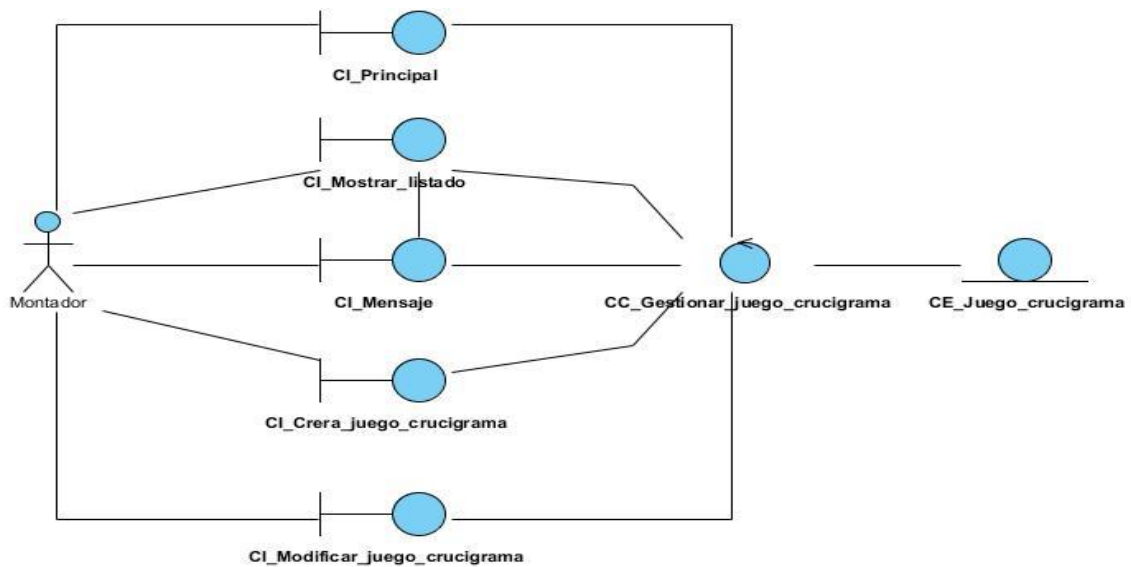


Ilustración 13. Gestionar juego Crucigrama.

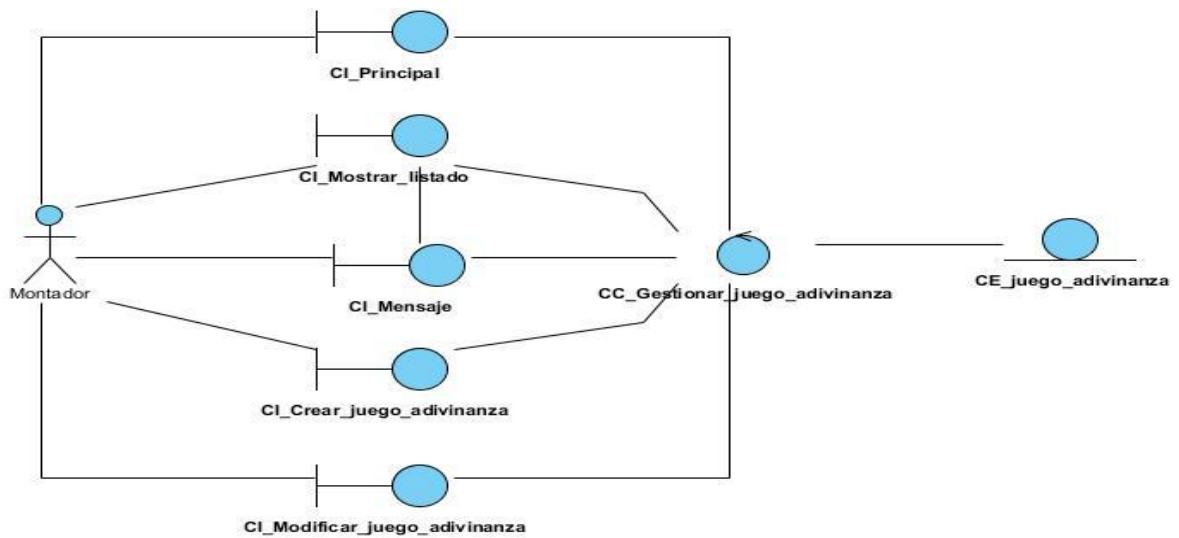


Ilustración 14. DCA_CU Gestionar juego Adivinanza.

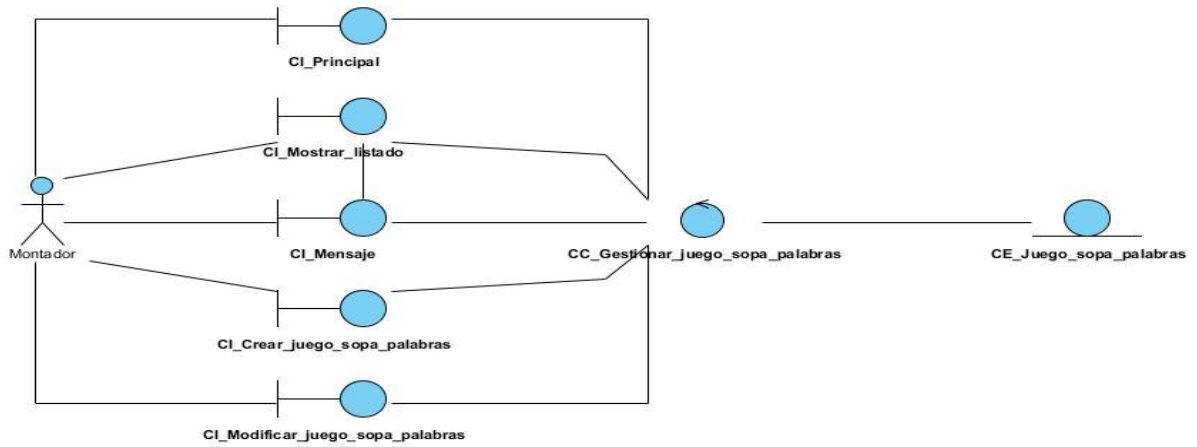


Ilustración 15. DCA_CU Gestionar juego Sopa de palabras.

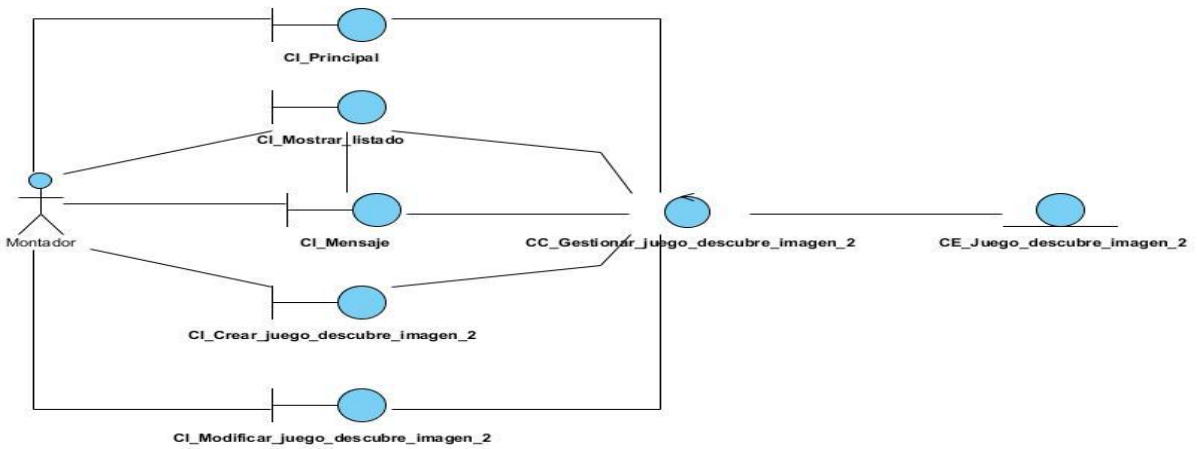


Ilustración 16. DCA_CU Gestionar juego Descubre la imagen 2.

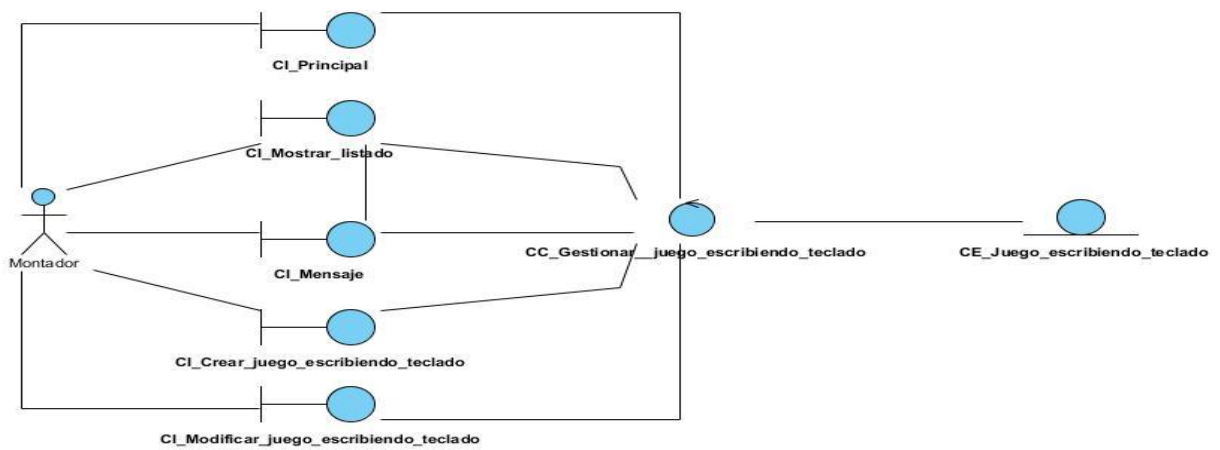


Ilustración 17. DCA_CU Gestionar juego Escribiendo con el teclado.

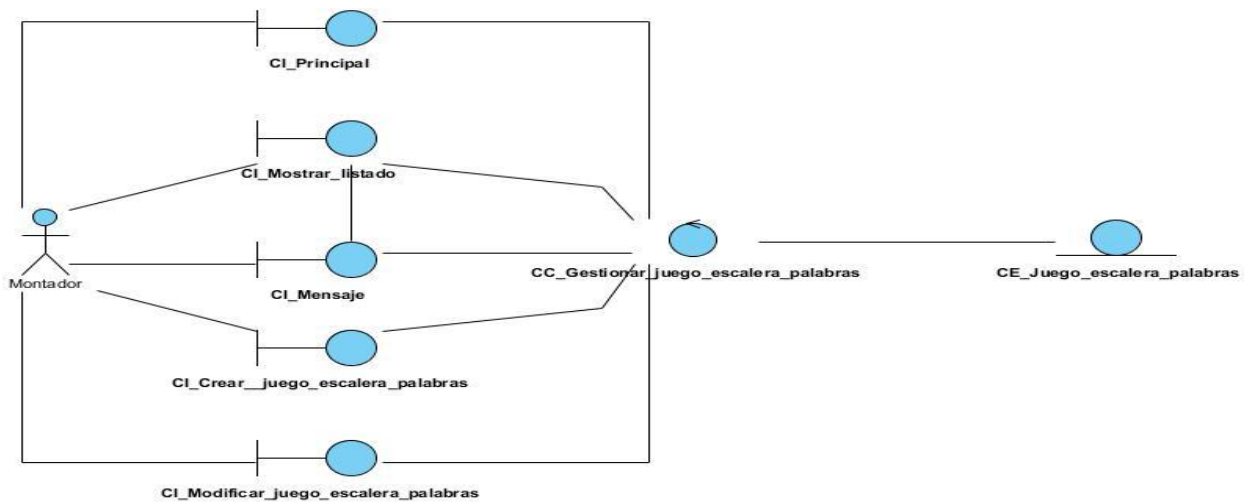


Ilustración 18. DCA_CU Gestionar juego Escalera de palabras.

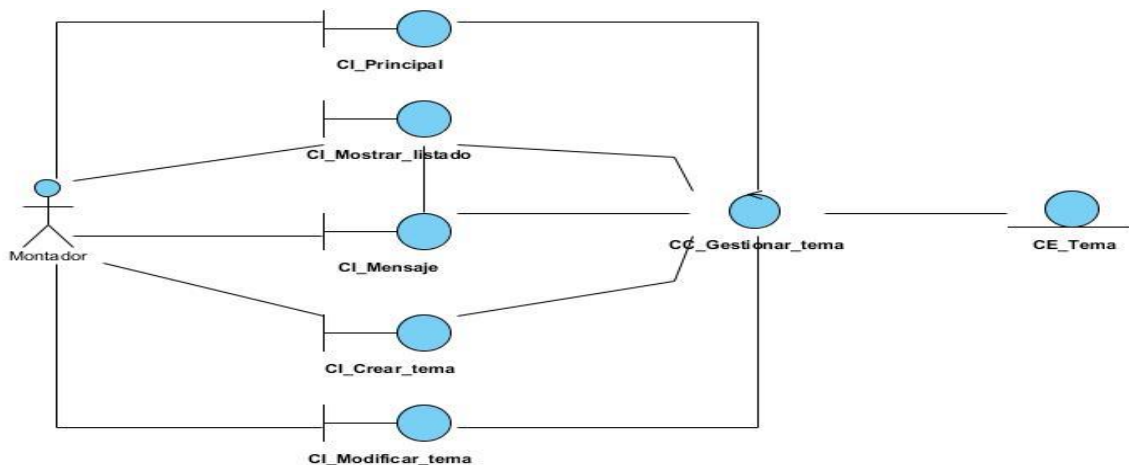


Ilustración 19. DCA_CU Gestionar tema.

3.2.2 Diagramas de Colaboración

Los diagramas de colaboración se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento.

A continuación se muestran los diagramas de colaboración más significativos. Los restantes diagramas se muestran en el Anexo 6.

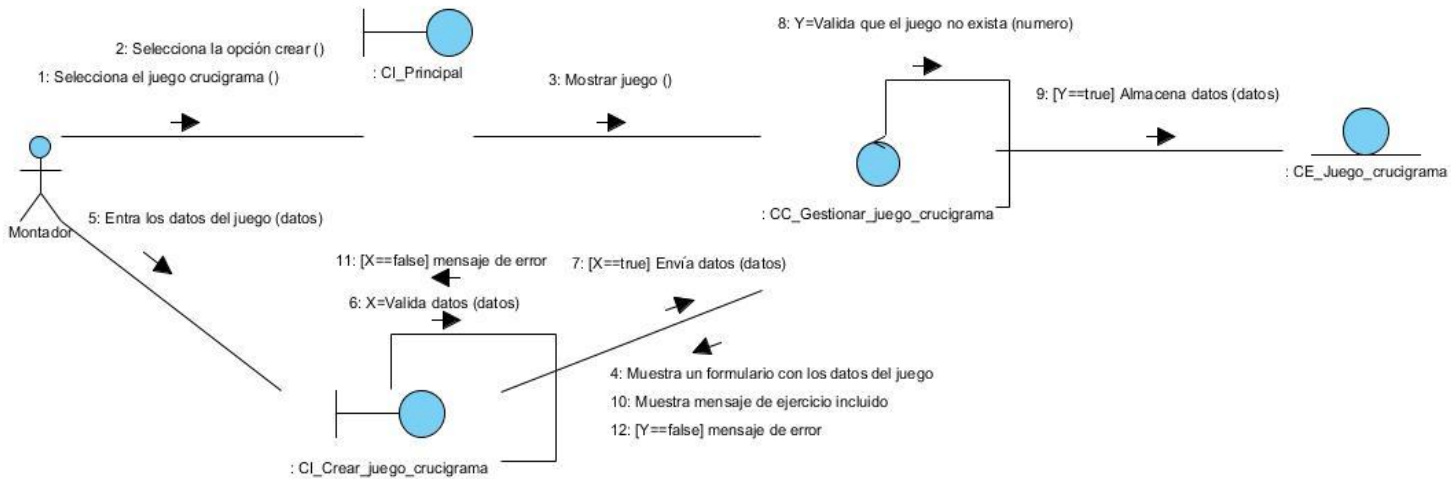


Ilustración 20. DC_CU Gestionar juego Crucigrama (Escenario Crear).

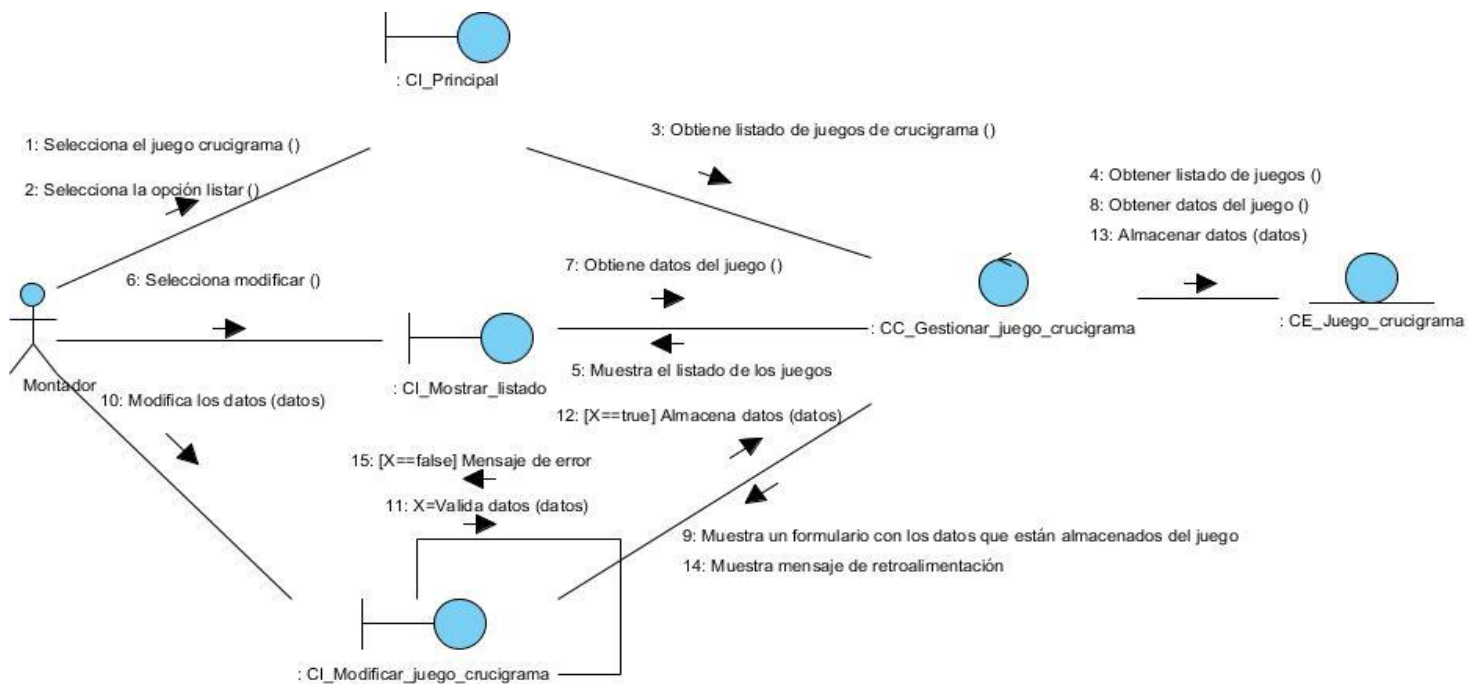


Ilustración 21. DC_CU Gestionar juego Crucigrama (Escenario Modificar).

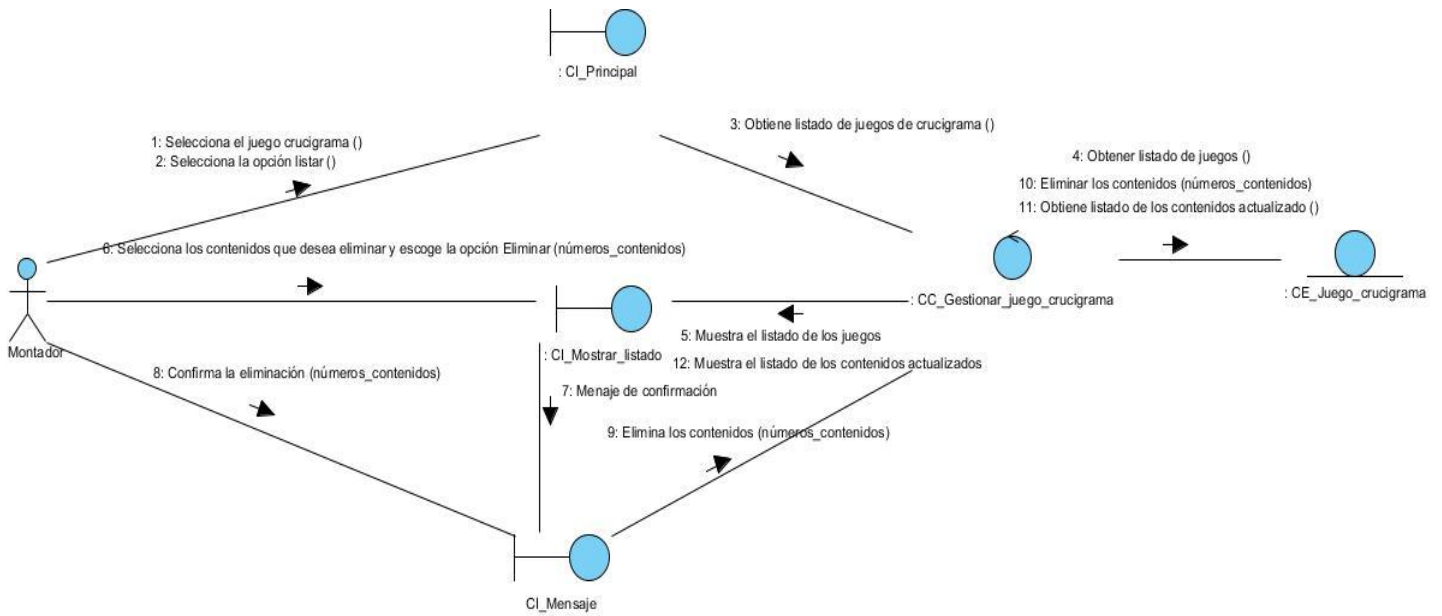


Ilustración 22. DC_CU Eliminar selección de contenido de juegos.

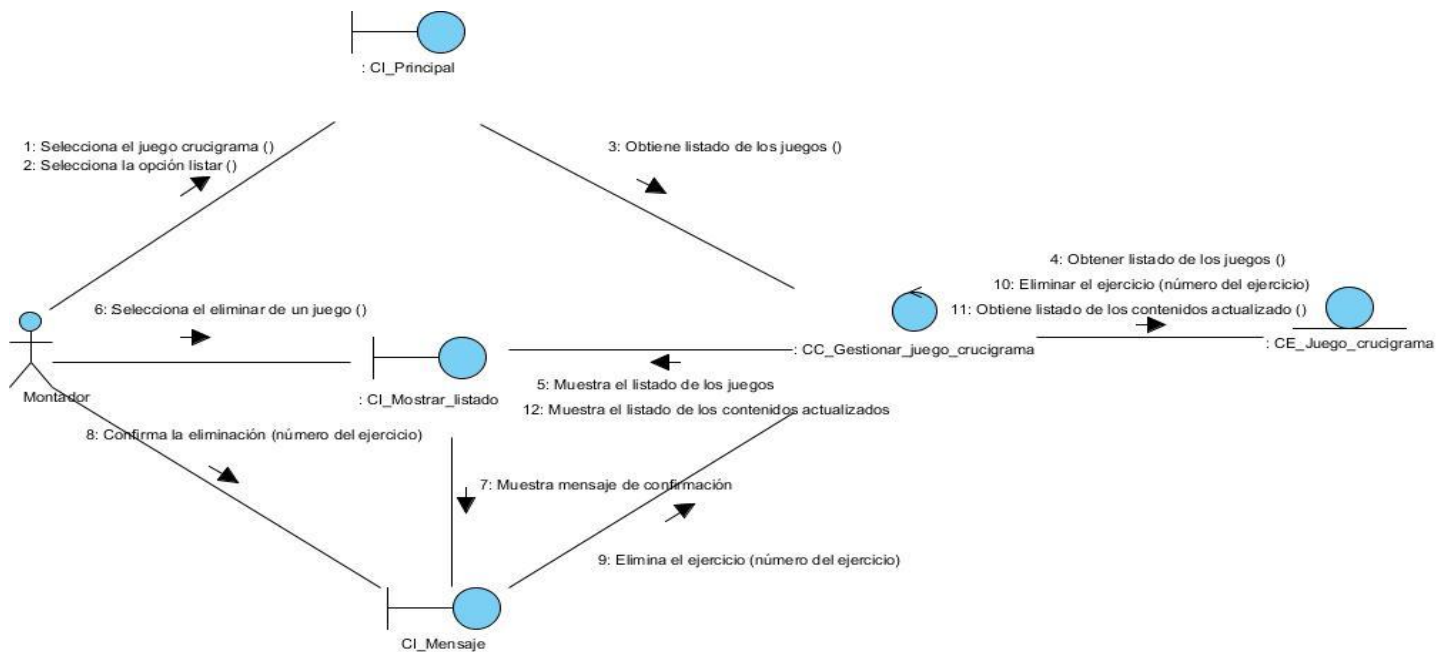


Ilustración 23. DC_CU Gestionar juego Crucigrama (Escenario Eliminar un contenido de juego).

3.3 Modelo de Diseño

El diseño radica en refinar el modelo de análisis para crear especificaciones adicionales que lo enriquecen con detalles próximos a la implementación. El propósito en el diseño es formular los modelos que se concentran en los requisitos no funcionales y en el dominio de la solución, por lo que es un modelo físico, no genérico, específico para una implementación. También descompone los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.

3.3.1 Patrones de Diseño

Son descripciones de objetos y clases que se comunican y que son capaces de solucionar un problema de diseño en general, en un contexto en particular.

Algunos patrones ayudan a estructurar un sistema en subsistemas base, otros a afinamiento de subsistemas y componentes, o las relaciones entre estos y algunos ayudan en la implementación en aspectos de diseño particulares en un lenguaje de programación específico.

Los patrones de diseño utilizados son los siguientes:

Objeto de Acceso a Datos (DAO, por sus siglas en inglés): DAO se utiliza para abstraer y encapsular todos los accesos a la fuente de datos, oculta completamente los detalles de implementación de la fuente de datos a sus clientes. El DAO maneja la conexión con la fuente de datos para obtener y almacenar datos. El DAO implementa el mecanismo de acceso requerido para trabajar con la fuente de datos. Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los mismos o de cuál es la fuente de almacenamiento.

Fachada (Facade): Es de tipo estructura. Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema. Encapsula los pasos necesarios para llevar a cabo lo que el cliente desea hacer. Ofrece un punto de acceso al resto de las clases, si las clases se sustituyen por otras o cambian, solo hay que actualizar la clase Fachada sin que el cambio afecte a las aplicaciones cliente.

Instancia única (Singleton): Es de tipo creación. Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia, es decir restringe la instanciación de una clase o valor de un tipo a un solo objeto.

Algunas de sus características son:

- El acceso a la instancia única está más controlado. Puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia.
- Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.
- Permite un número variable de instancias.

3.3.2 Diagrama de Clases del Diseño

Uno de los artefactos más importantes a lograr en el flujo de trabajo Diseño es el diagrama de clases del diseño, donde se exponen las clases que intervienen en las realizaciones de los casos de uso del sistema. En este tipo de diagrama se representa un nivel de detalle más alto que en el diagrama de clases del análisis, relacionándose con el lenguaje de programación del cual se hará uso en la implementación del sistema. Un diagrama de clases del diseño es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Estereotipo



<<Client Page >>



<<Form>>



<<Server Page>>

Explicación

Representa una página Web, con formato HTML. Son interpretadas por el navegador, su función es visualizar, interactuar y mostrar lo que el usuario necesita.

Colección de elementos de entrada que son parte de una página cliente. Su función es enviar los datos entrados por el usuario a la página servidora.

Representa la página Web que tiene código que se ejecuta en el servidor. Su función es construir la página cliente.

A continuación se presentan uno de los diagramas de clases del diseño. Los restantes diagramas se muestran en el Anexo 7.

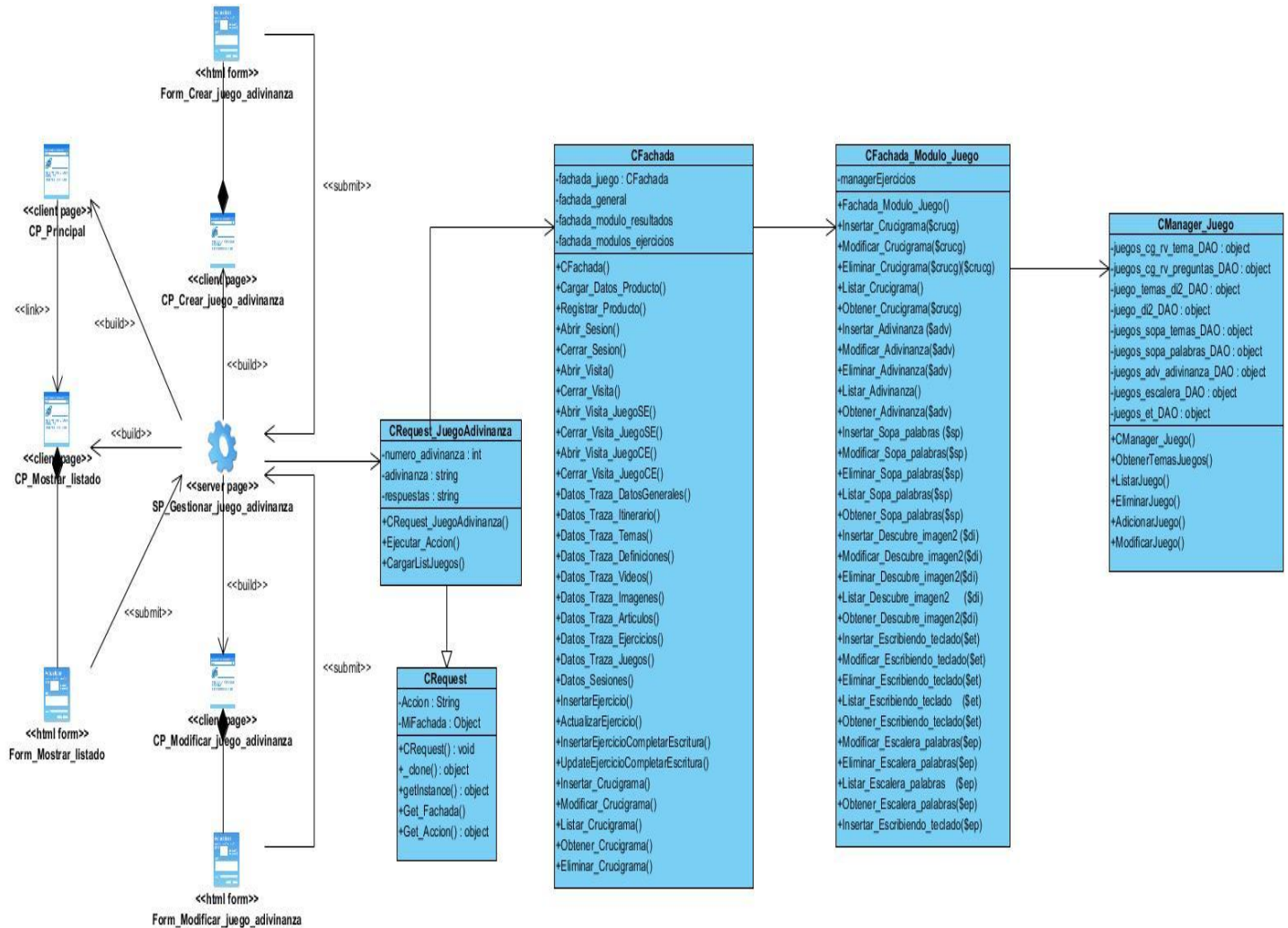


Ilustración 24. DCD_CU Gestionar juego Adivinanza.

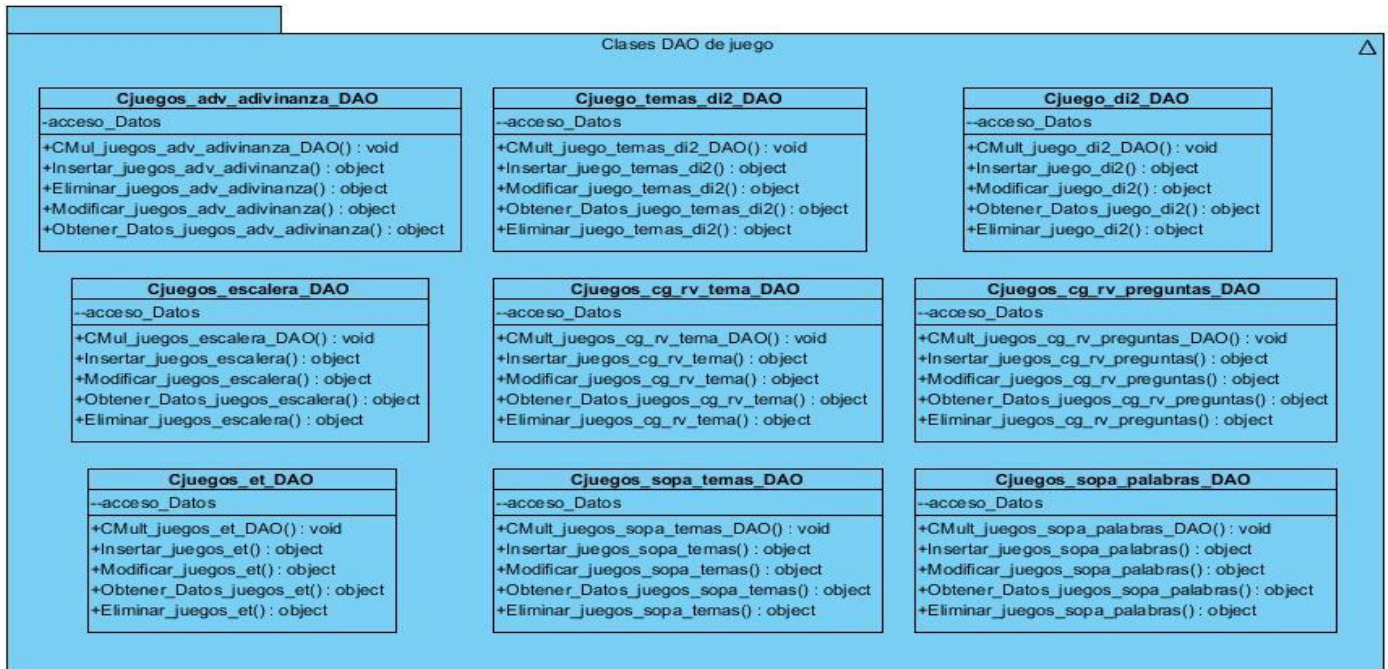


Ilustración 25. Paquete de clases DAO de juego.

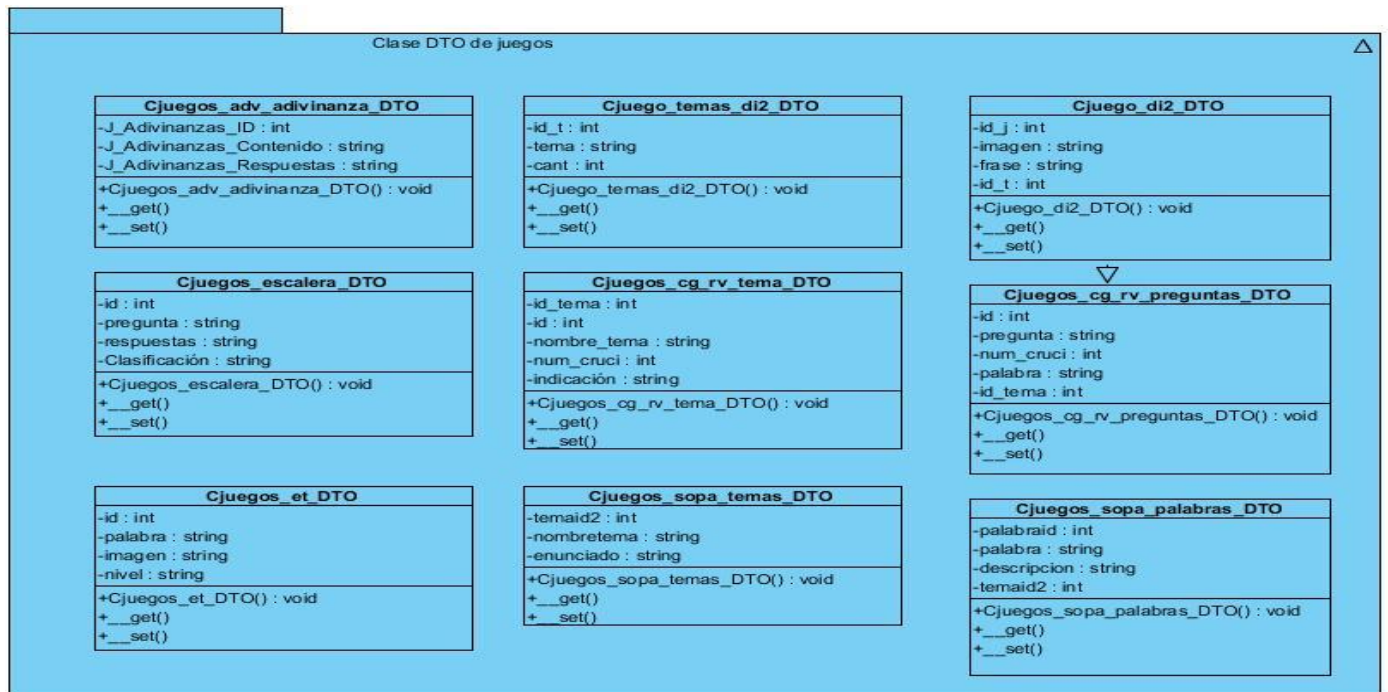


Ilustración 26. Paquete de clases DTO de juego.

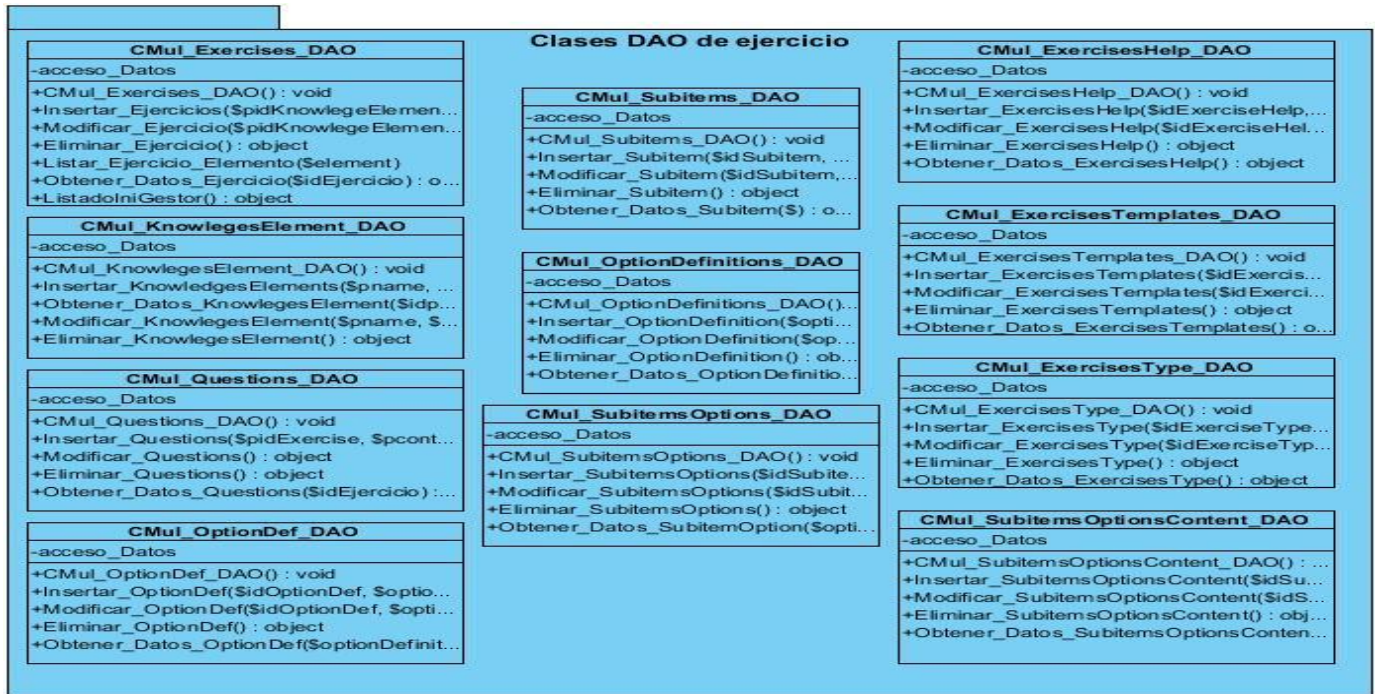


Ilustración 27. Paquete de clases DAO de ejercicio.

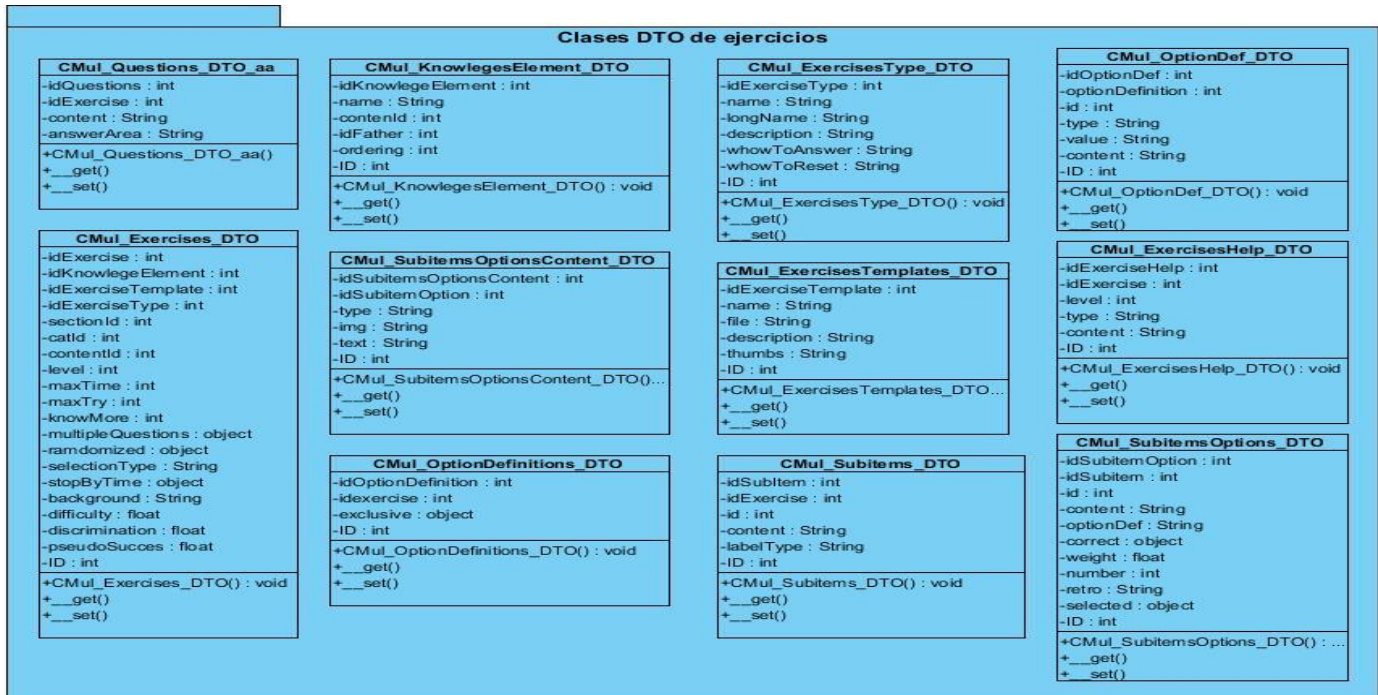


Ilustración 28. Paquete de clases DTO de ejercicio.

3.4 Modelo de Datos

Un modelo de datos es un lenguaje orientado a representar una base de datos. Este permite describir el tipo de los datos que hay en la base y la forma en que se relacionan.

En la Figura 42 y 43 se muestra el modelo de datos correspondiente al módulo ejercicios y juegos respectivamente, de la versión 1.1 de la herramienta GCon para gestionar los contenidos de la colección La Caja Mágica.

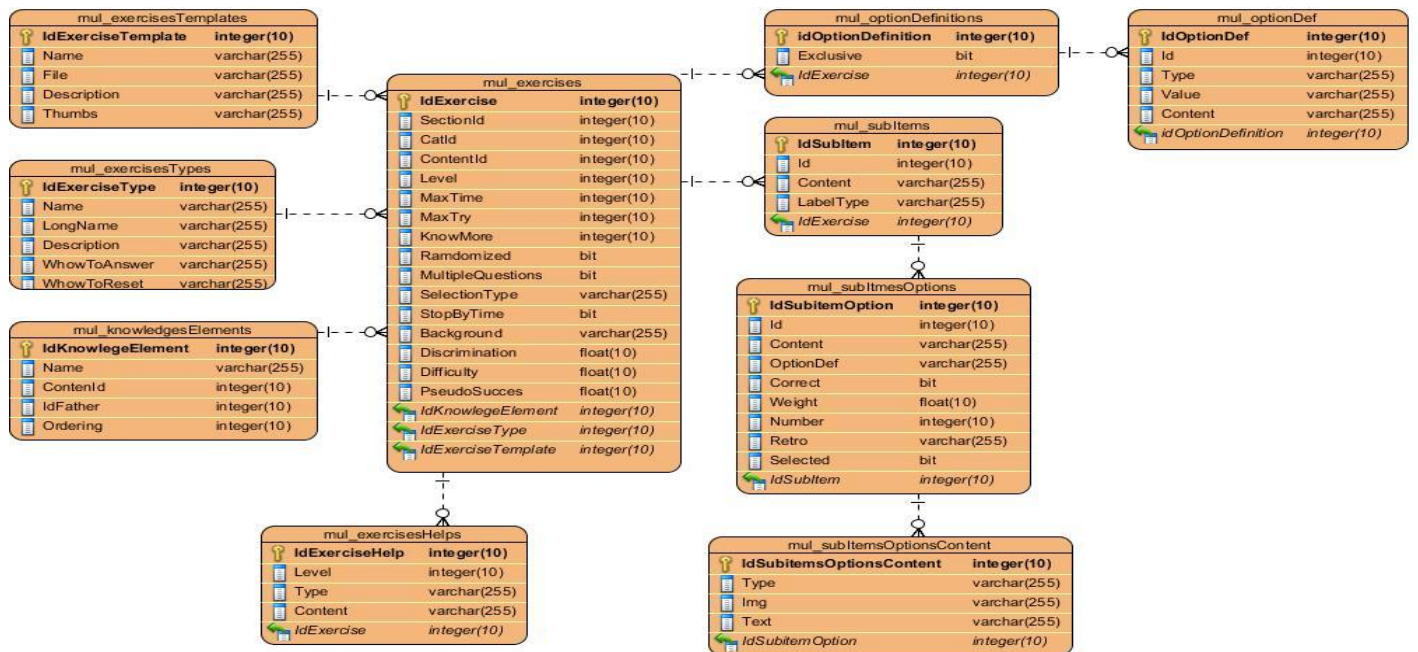


Ilustración 29. Modelo de datos de ejercicios.

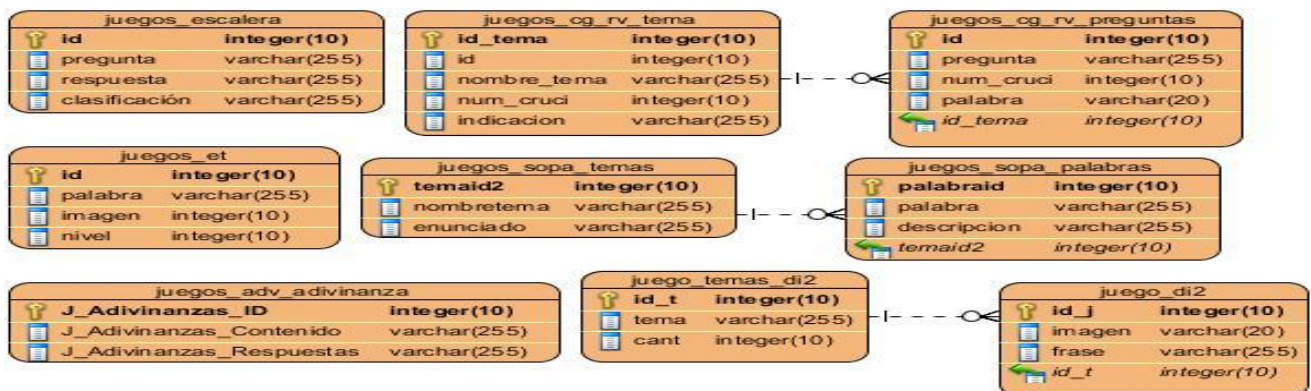


Ilustración 30. Modelo de datos de juegos.

3.5 Conclusiones

En este capítulo se ha diseñado el sistema a través de una serie de modelos que serán entregados a los implementadores, facilitándoles las funcionalidades a implementar con el fin de obtener una aplicación que contenga los requisitos que desea el cliente. Para lograr un mejor diseño se utilizaron algunos patrones que ayudan a la optimización del mismo. Además, se realizó el diseño de la base de datos, generando el modelo físico de datos.

Conclusiones generales

Luego de la realización de este trabajo se puede arribar a las siguientes conclusiones:

- Se alcanzó un mejor entendimiento de las características del sistema, mediante la realización del modelo de dominio, la identificación de los requisitos funcionales y no funcionales, así como la elaboración del diagrama de casos de uso del sistema y la descripción detallada de cada uno de estos.
- A través del análisis y diseño del sistema se obtuvo la documentación necesaria que servirá como punto de partida para la etapa de implementación.

Recomendaciones

Para un futuro desarrollo de la herramienta GCon se recomienda:

- Dar seguimiento al presente trabajo con la implementación y siguientes flujos de trabajo que propone la metodología utilizada.
- Tener en cuenta el trabajo como material de apoyo para futuras investigaciones.

Referencias bibliográficas

1. Colectivos de autores. Preparación pedagógica integral para profesores universitarios. Ciudad de La Habana: Félix Varela, 2003.
2. Ramos Pérez L, Domínguez Lovaina J, Gavilondo Mariño X, Fresno Chávez C. ¿Software educativo, hipermedia o entorno educativo? Ciudad de La Habana, Cuba: Centro Nacional de Información de Ciencias Médicas-Infomed, 2008. Available from: http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm.
3. LANZ C. Cuatro líneas estratégicas del Proyecto Educativo Nacional: Concreción de la Pedagogía Alternativa. S.l.: s.n.
4. Multimedia. In: [online]. [Cited: enero 17, 2012]. Available from: <http://es.tech-faq.com/multimedia.html>.
5. La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual-Archivo de artículos del Observatorio para la CiberSociedad. In: [online]. [Cited: enero 17, 2012]. Available from: <http://www.cibersociedad.net/archivo/articulo.php?art=218>.
6. MSc. Rodríguez, Mario del Toro. Modelos de diseño didáctico de hiperentornos de enseñanza-aprendizaje. Resumen de la Tesis en opción al grado de Doctor en Ciencias Pedagógicas. Ciudad de La Habana, República de Cuba: Instituto Superior Pedagógico "Enrique José Varona", 2006.
7. Dr. C. Iván Barreto Gelles, Dr. C. Mario del Toro Rodríguez and MSc. Aniuska Riquenes Tarragó. *tecnología educativa: dos modelos para la acción del maestro*. Dr. C. María Julia Moreno Castañeda. Cuba: s.n., 2009. ISBN 978-959-18-0430-3.
8. Ardora. Creación de actividades escolares. In: [online]. [Cited: enero 18, 2012]. Available from: http://www.webardora.net/index_cas.htm.
9. Hot Potatoes Home Page. In: [online]. [Cited: enero 18, 2012]. Available from:

- <http://hotpot.uvic.ca/index.php#news>.
10. CRODA. In: [online]. [Cited: enero 18, 2012]. Available from: <http://croda.uci.cu/>.
 11. Herramientas de autor: ExeLearning | Suite101.net. In: [online]. [Cited: enero 18, 2012]. Available from: <http://ana-maria-villar-varela.suite101.net/herramientas-de-autor-ExeLearning-a39148>.
 12. Jacobson, Booch y Rumbaugh. El Proceso Unificado de Desarrollo de Software.
 13. Rumbaugh, Jacobson y Booch. El lenguaje unificado de modelado.
 14. Free Download Manager. *Free Download Manager*. In: [online]. [Cited: enero 20, 2012]. Available from:
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
 15. Los diferentes lenguajes de programación para la web. In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
 16. Usabilidad y arquitectura del software. In: [online]. [Cited: febrero 3, 2012]. Available from: <http://www.desarrolloweb.com/articulos/1622.php>.
 17. www.petra.euitio.uniovi.es. www.petra.euitio.uniovi.es. [online]. [Cited: enero 21, 2012]. Available from: [http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos de Desarrollo Integrado.pdf](http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf).
 18. Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto. In: [online]. [Cited: febrero 10, 2012]. Available from: http://netbeans.org/index_es.html.
 19. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? In: [online]. [Cited: febrero 15, 2012]. Available from: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.

20. MySQL:: About MySQL. In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.mysql.com/about/>.
21. ¿Qué es un servidor? - Definición y explicación del término servidor. In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.masadelante.com/faqs/servidor>.
22. Apache. *Apache*. [online]. [Cited: febrero 2, 2012]. Available from: http://httpd.apache.org/docs/2.0/es/new_features_2_0.html.

Bibliografía

1. Colectivos de autores. Preparación pedagógica integral para profesores universitarios. Ciudad de La Habana: Félix Varela, 2003.
2. Ramos Pérez L, Domínguez Lovaina J, Gavilondo Mariño X, Fresno Chávez C. ¿Software educativo, hipermedia o entorno educativo? Ciudad de La Habana, Cuba: Centro Nacional de Información de Ciencias Médicas-Infomed, 2008. Available from: http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm.
3. LANZ C. Cuatro líneas estratégicas del Proyecto Educativo Nacional: Concreción de la Pedagogía Alternativa. S.l.: s.n.
4. Multimedia. In: [online]. [Cited: enero 17, 2012]. Available from: <http://es.tech-faq.com/multimedia.html>.
5. La Tecnología de la Información y la Comunicación (TIC). Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual-Archivo de artículos del Observatorio para la CiberSociedad. In: [online]. [Cited: enero 17, 2012]. Available from: <http://www.cibersociedad.net/archivo/articulo.php?art=218>.
6. MsC. Rodríguez, Mario del Toro. Modelos de diseño didáctico de hiperentornos de enseñanza-aprendizaje. Resumen de la Tesis en opción al grado de Doctor en Ciencias Pedagógicas. Ciudad de La Habana, República de Cuba: Instituto Superior Pedagógico "Enrique José Varona", 2006.
7. DR. C. Iván Barreto Gelles, DR. C. Mario del Toro Rodríguez and MSc. Aniuska Riquenes Tarragó. *Tecnología Educativa: Dos modelos para la acción del maestro*. Dr. C. María Julia Moreno Castañeda. Cuba: s.n., 2009. ISBN 978-959-18-0430-3.
8. Ardora. Creación de actividades escolares. In: [online]. [Cited: enero 18, 2012]. Available from: http://www.webardora.net/index_cas.htm.
9. Hot Potatoes Home Page. In: [online]. [Cited: enero 18, 2012]. Available from:

- <http://hotpot.uvic.ca/index.php#news>.
10. CRODA. In: [online]. [Cited: enero 18, 2012]. Available from: <http://croda.uci.cu/>.
 11. Herramientas de autor: ExeLearning | Suite101.net. In: [online]. [Cited: enero 18, 2012]. Available from: <http://ana-maria-villar-varela.suite101.net/herramientas-de-autor-ExeLearning-a39148>.
 12. CyTA. In: [online]. [Cited: enero 19, 2012]. Available from: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
 13. Agile Project Management with Scrum-Ken Schwaber-Google Libros. In: [online]. [Cited: enero 19, 2012]. Available from: http://books.google.es/books?hl=es&lr=&id=RpYX01XVMksC&oi=fnd&pg=PT1&dq=SCRUM&ots=IdpBeskBGO&sig=LIGDMb_Q4AbWT6By8dB952JUvUk#v=onepage&q=SCRUM&f=false.
 14. Jacobson, Booch y Rumbaugh. El Proceso Unificado de Desarrollo de Software.
 15. Los diferentes lenguajes de programación para la web. In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
 16. Rumbaugh, Jacobson y Booch. El lenguaje unificado de modelado.
 17. Rosegrady Booch. Visual Modeling with Rational Rose 2000 and UML. S.I. s.n., [Cited: febrero 3, 2012].
 18. UML Modeler-UML, ERD, Requirements Diagram. In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.visual-paradigm.com/product/vpuml/editions/modeler.jsp>.
 19. Usabilidad y arquitectura del software. In: [online]. [Cited: febrero 3, 2012]. Available from: <http://www.desarrolloweb.com/articulos/1622.php>. PostgreSQL: About. In: [online]. [Cited: enero 22, 2012]. Available from: <http://www.postgresql.org/about/>.
 20. www.petra.euitio.uniovi.es. www.petra.euitio.uniovi.es. [online]. [Cited: enero 21, 2012]. Available

- from: <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos de Desarrollo Integrado.pdf>
21. Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto. In: [online]. [Cited: febrero 10, 2012]. Available from: http://netbeans.org/index_es.html.
 22. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? In: [online]. [Cited: febrero 15, 2012]. Available from: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
 23. PostgreSQL: About. In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.postgresql.org/about/>.
 24. MySQL:: About MySQL. In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.mysql.com/about/>.
 25. Tutorial Qué es Oracle? [online-emagister.com](http://www.emagister.com). In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.emagister.com/que-es-oracle-cursos-1053949.htm#programa>.
 26. ¿Qué es un servidor? - Definición y explicación del término servidor. In: [online]. [Cited: enero 20, 2012]. Available from: <http://www.masadelante.com/faqs/servidor>.
 27. Apache. *Apache*. [online]. [Cited: febrero 2, 2012]. Available from: http://httpd.apache.org/docs/2.0/es/new_features_2_0.html.
 28. PRESSMAN, Roger. Ingeniería del Software. Un enfoque práctico. s.l.: Interamericana, 2002.
 29. Smaldone, Javier. Desarrollo de Software [online] 2010. [Cited: febrero 23, 2012]. Available from: <http://www.smaldone.com.ar/profesional/programacion.shtml>.
 30. Sitio oficial de PHP. [online] 2010. [Cited: diciembre 12, 2011]. Available from: <http://php.net/>.
 31. Sitio oficial de Visual Paradigm. [online] 2010. [Cited: febrero 25, 2012]. Available from: <http://www.visual-paradigm.com/>.

32. LARMAN, Craig. "UML y Patrones". s.l.: Prentice Hall, 2000. ISBN-84-205-3438-2.
33. Sitio oficial de PostgreSQL. [online] 2010. [Cited: enero 22, 2012]. Available from: <http://www.postgresql.org/about/>.
34. Lago, Ramiro. Patrones de diseño de software. [online] 2007. [Cited: febrero 13, 2012]. Available from: <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
35. Dr. Marquéz, Pere. Multimedia educativo: clasificación, funciones, ventajas e inconvenientes. [online] enero 17, 2009. [Cited: enero 20, 2012]. Available from: <http://www.pangea.org/peremarques/funcion.htm>.
36. La Informática educativa en el contexto actual. Rodríguez Lamas, Raúl. s.l.: Edutec. Revista Electrónica de Tecnología Educativa, 2000. Available from: <http://www.uib.es/depart/gte/edutec-e/Revelec13/Rlamas.html>.
37. EGUÍLUZ Pérez, Javier. Introducción a JavaScript. 2008. [online]. [Cited: enero 29, 2012.]. Available from: <http://es.scribd.com/hbalmes/d/65108248-Introduccion-Javascript-Javier-Eguiluz-Perez>
38. PHP. In: [online]. [Cited: Enero 29, 2012.] Available from: <http://www.php.net/manual/es/getting-started.php>
39. Jumbo, Fabián. Multimedia educativa. [online] 2008. Available from: <http://www.slideshare.net/jumbofabian/multimedia-educativa-presentation-715361>.
40. Núñez Rojas, Nemecio. Concepción del Software educativo. La Webquest, el aula virtual y el desarrollo de competencias para la investigación en los estudiantes del I Ciclo de educación – USAT. 2010. Available from: www.eumed.net/libros/2010a/669/.
41. Araujo Pérez , Liana Isabel. *Plan de Desarrollo de Software v2.0*. La Habana : s.n., 2007.

42. Una Introducción a Apache. In: [online]. [Cited: enero 23, 2012]. Available from: http://linux.ciberaula.com/articulo/linux_apache_intro/.
43. López Pecho, Ramiro y Ballesteros, Julio César. HERRAMIENTAS CASE. *HERRAMIENTAS CASE*. [online] 29 de Septiembre de 2008. [Cited: febrero 1, 2012]. Available from: <http://tpsis324.blogspot.com/>.

Glosario de términos

TIC: Se denominan Tecnologías de la Información y las Comunicaciones (TIC), al conjunto de Tecnologías que permiten la adquisición, producción, almacenamiento, tratamiento, comunicación, registro y presentación de informaciones.

Software educativo: Software destinado a la enseñanza - aprendizaje, combina recursos multimedia para mostrar la información como: imágenes, sonidos, videos y animaciones.

Multimedia: Tecnología que utiliza múltiples medios de expresión para mostrar información como: textos, imágenes, sonidos y videos.

Hipertexto: Organización de información en diferentes nodos, conectados entre sí a través de enlaces.

Hipermedia: Tecnología que permite estructurar la información de una manera no lineal, a través de nodos interconectados por enlaces. La información presentada en estos nodos podrá integrar diferentes medios tales como: texto, sonido, gráficos y video.

mp3: Formato de compresión de archivos audio. Posibilita almacenar música de alta calidad ocupando un mínimo espacio.

FLV: Flash Video, es el formato de video de Macromedia para transmisión de video digital por Internet.

SCORM: Shareable Content Object Reference Model, es un conjunto de especificaciones para desarrollo, empaquetamiento y distribución de material educativo.

LOM: Learning Object Metadata, en español Metadatos para Objetos de Aprendizaje, es el estándar de metadatos para objetos de aprendizaje.

E-learning: Enseñanza a distancia caracterizada por una separación física entre profesores y alumnos, donde se usa preferiblemente Internet como medio de comunicación y de distribución del conocimiento.

Sistema Operativo: programa informático que actúa de interfaz entre los dispositivos de hardware y los programas usados por el usuario para manejar la computadora.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

XSS: XSS, del inglés Cross-site Scripting es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado. Las vulnerabilidades de XSS ocurren cuando un atacante utiliza una aplicación web para mandar código malicioso, generalmente en la forma de un script, a un usuario diferente.

API: Application Programming Interface, en español Interfaz de Programación de Aplicaciones, es un conjunto de funciones residentes en bibliotecas que permiten que una aplicación corra bajo un determinado sistema operativo.

DCA_CU: Diagrama de Clases del Análisis del Caso de Uso.

DCD_CU: Diagrama de Clases del Diseño del Caso de Uso.

Anexos

Anexo 1: Representación de Hipermedia



Ilustración 31. Representación de hipermedia.

Anexo 2: Disciplinas, fases e iteraciones de RUP

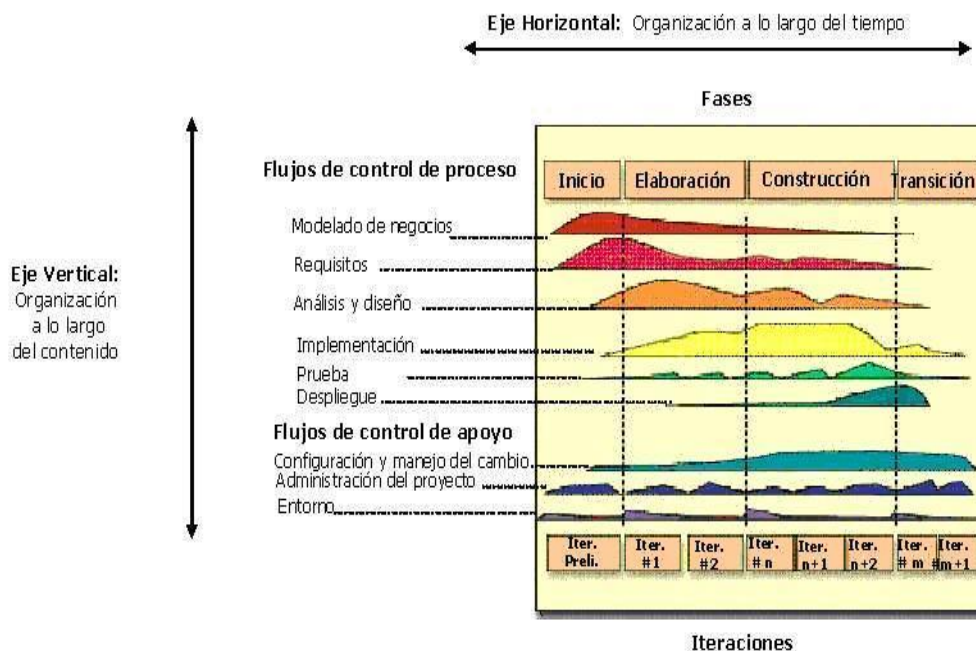


Ilustración 32. Disciplinas, fases e iteraciones de RUP.

Anexo 3: Arquitectura 3 Capas

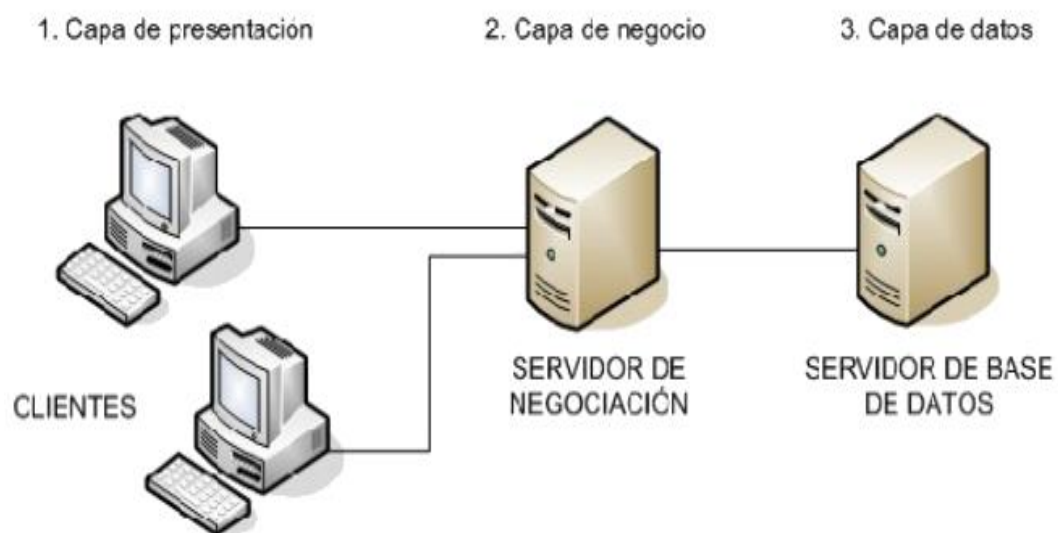


Ilustración 33. Arquitectura 3 Capas.