



Facultad 4

# *Versión 2.0 de la Colección de Juegos Educativos*

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

**Autores:** Mayara Portales Alvarez

Jorge Remberto Rodríguez Rodríguez

**Tutores:** Lic. Lesyanis Pompa Arcia

Ing. Katia Pardo Duarte

Ing. Rafael Jacobo Hidalgo Urbino

Ing. Osmany Montes de Oca Rodríguez

La Habana, Junio de 2012

“Año 54 de la Revolución”

Declaramos ser los autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas para que haga uso de la misma, como estime pertinente, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los    días del mes de            del año    .

**Mayara Portales Alvarez**

**Jorge Remberto Rodríguez Rodríguez**

\_\_\_\_\_

\_\_\_\_\_

Firma del Autor

Firma del Autor

Lesyanis Pompa Arcia

Katia Pardo Duarte

\_\_\_\_\_

\_\_\_\_\_

Firma del Tutor

Firma del Tutor

Osmany Montes de Oca Rodríguez

Rafael Jacobo Hidalgo Urbino

\_\_\_\_\_

\_\_\_\_\_

Firma del Tutor

Firma del Tutor

## PENSAMIENTO

*"Hay una fuerza motriz más  
poderosa que el vapor, la  
electricidad y la energía atómica:  
la voluntad."*

Albert Einstein

## AGRADECIMIENTOS

*A mi madre, por darme todo su amor, confianza y apoyo en estos 23 años de vida; por ser mi inspiración en cada momento y estar siempre conmigo, no importa a donde vaya.*

*A mi familia, por el apoyo que siempre me ha dado, principalmente en estos 5 años de la carrera.*

*A mi flaca, por preservar esta amistad, aun cuando la distancia se impone, por estar siempre disponible cuando la necesito y aguantar cada una de mis pesadeces.*

*A mis compañeros y amigos de la FEU, en especial Olia, Linet y Yunion, por todos esos buenos y malos momentos, de estrés y alegría, de trabajo y sacrificio que pasamos juntos, nunca los olvidaré.*

*A mis amigos y amigas Lili, Lila, Yuni, Osmary, Damari, Berto, Frank, Ariel, Mirelio, Mandy, Yanet, Ernesto y muchos más que no me alcanzaría la hoja para ponerlos.*

*A mi novio Henry, por el super apoyo que me ha dado en este tiempito que llevamos juntos, por no dejar que me desanimara en ningún momento y por todos esos buenos consejos y charlas educativas que siempre me da.*

*A todos los profesores que me impartieron clases, en especial a los del proyecto Multisaber-Navegante con los cuales compartí durante 2 años: Liana, Osmany, Katia, Ismael, Ana, Rosalba, Osdalme, Hector y Jacobo.*

*A todas las personas que aportaron su granito de arena para que hoy pudiera cumplir mi sueño.*

*A todos muchas gracias.*

**Mayara**

## AGRADECIMIENTOS

*A mis padres por permitirme ser quien soy.*

*A mi abuela María por su apoyo incondicional a lo largo de toda mi vida.*

*A mi hermanito por estar cerca de mis padres y ocupar el lugar añorado por mí durante estos años de estudio.*

*A mi familia por su constancia y por su ejemplo en la construcción de mi proyecto de vida.*

*A mi mujer y a mi retoño de vida, por todos los momentos de felicidad y por aquellos difíciles en los que siempre hemos permanecido unidos, por todo el amor y la inspiración que recibo estando junto a ustedes.*

*A la dirección de la facultad y colectivo de profesores por el respeto que me inspiran y por su aporte en mi formación como profesional.*

*A mis profesoras y profesores Liana, Katia, Osmany, Jacobo, Jorge, Hector y Osdalme por ofrecerme desde el aula y el proyecto la sabiduría necesaria para mi desempeño como Ingeniero Informático.*

*A todos los amigos del proyecto Multisaber Navegante y de las comunidades UCA por compartir experiencias y estar a mi lado durante estos últimos tres años.*

*A todas las personas que han exigido y contribuido a que sea mejor ser humano mis agradecimientos.*

**Jorge Remberto**

## DEDICATORIA

*A mi madre, la mejor  
de todas.*

**MAYARA**

*A mi pequeña hija, a  
mis queridos padres y  
a mi abuela María.*

**JORGE REMBERTO**

## RESUMEN

### **Resumen**

En la presente tesis que tiene por título Versión 2.0 de la Colección de Juegos Educativos, se exponen los resultados de una investigación que tuvo por objetivo lograr una colección de juegos educativos multiplataforma, estable y que permita la gestión de los contenidos. Partiendo de un enfoque dialéctico materialista fueron utilizados diversos métodos teóricos y empíricos; los que permitieron estudiar el desarrollo del software educativo a nivel internacional, en Cuba y en la Universidad de las Ciencias Informáticas, así como la determinación de la metodología (Rational Unified Process), herramientas (Visual Paradigm, Netbeans IDE), lenguajes (JavaScript, PHP) y frameworks (Dojotoolkit, Codeigniter) necesarios para el desarrollo de la propuesta. La estructura del trabajo se concibió a partir de la definición de cuatro capítulos. El aporte práctico de esta tesis, lo constituye la implementación de una aplicación dirigida a la enseñanza primaria, la cual apoya el proceso de enseñanza-aprendizaje, a la vez que le proporciona al profesor la posibilidad de crear nuevos contenidos y adaptarlos a las características de los estudiantes. Para la realización de este trabajo de diploma se consultó bibliografía actualizada permitiendo la constatación de información reciente del objeto de estudio en cuestión. Finalmente se efectuó el análisis, diseño e implementación como resultado de los requerimientos establecidos para garantizar el cumplimiento del objetivo general.

# ÍNDICE DE CONTENIDO

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....</b>	<b>5</b>
1.1 INTRODUCCIÓN.....	5
1.2 ANÁLISIS DE SOLUCIONES SIMILARES EXISTENTES.....	5
1.3 METODOLOGÍAS DE DESARROLLO.....	14
1.4 LENGUAJE UNIFICADO DE MODELADO.....	17
1.5 HERRAMIENTAS CASE.....	18
1.6 LENGUAJES DE DESARROLLO.....	19
1.7 HERRAMIENTAS DE DESARROLLO.....	22
1.8 FRAMEWORK.....	23
1.9 GESTOR DE BASE DE DATOS.....	26
1.10 SERVIDOR WEB.....	28
1.11 CONCLUSIONES.....	29
<b>CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....</b>	<b>30</b>
2.1 INTRODUCCIÓN.....	30
2.2 MODELO DE DOMINIO.....	30
2.3 ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA.....	31
2.4 MODELO DEL SISTEMA.....	36
2.5 CONCLUSIONES.....	42
<b>CAPÍTULO 3. ANÁLISIS Y DISEÑO.....</b>	<b>43</b>
3.1 INTRODUCCIÓN.....	43
3.2 ¿QUÉ ES EL ANÁLISIS?.....	43
3.3 MODELO DE ANÁLISIS.....	43
3.4 PATRÓN ARQUITECTÓNICO.....	44
3.5 PATRONES DE DISEÑO.....	45
3.6 MODELO DE DISEÑO.....	47
3.7 DISEÑO DE LA BD.....	48
3.8 CONCLUSIONES.....	51
<b>CAPÍTULO 4. IMPLEMENTACIÓN.....</b>	<b>52</b>
4.1 INTRODUCCIÓN.....	52
4.2 MODELO DE IMPLEMENTACIÓN.....	52
4.3 TRATAMIENTO DE ERRORES.....	54
4.4 SEGURIDAD.....	55
4.5 CONCLUSIONES.....	55
<b>CONCLUSIONES GENERALES.....</b>	<b>56</b>
<b>RECOMENDACIONES.....</b>	<b>57</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>58</b>



# ÍNDICE DE FIGURAS

## Índice de Figuras

FIGURA 1: MODELO DEL DOMINIO.....	31
FIGURA 2: DIAGRAMA DE CASOS DE USO (FRONTEND).....	37
FIGURA 3: DIAGRAMA DE CASOS DE USO (BACKEND1).....	38
FIGURA 4: DIAGRAMA DE CASOS DE USO (BACKEND2).....	38
FIGURA 5: DIAGRAMA DE CASOS DE USO (BACKEND3).....	39
FIGURA 6: DIAGRAMA DE CLASES DEL ANÁLISIS CU GESTIONAR_EJERCICIOS.....	44
FIGURA 7: DIAGRAMA DE CLASES DE COLABORACIÓN CU REALIZAR_JUEGO (ESCENARIO JUEGO_EA).....	44
FIGURA 8: DIAGRAMA DE CLASES DE COLABORACIÓN CU AUTENTICAR_ADMIN.....	44
FIGURA 9: DIAGRAMA DE CLASES DEL DISEÑO CU REALIZAR_JUEGO.....	48
FIGURA 10: DIAGRAMA DE CLASES PERSISTENTES.....	49
FIGURA 11: DIAGRAMA ENTIDAD-RELACIÓN.....	50
FIGURA 12: DIAGRAMA DE DESPLIEGUE 1.....	52
FIGURA 13: DIAGRAMA DE DESPLIEGUE 2.....	53
FIGURA 14: DIAGRAMA DE COMPONENTES GENERAL.....	53
FIGURA 15: DIAGRAMA DE COMPONENTES DEL PAQUETE GAMEBOX.....	54
FIGURA 16: DIAGRAMA DE COMPONENTES DEL PAQUETE MODELO.....	54

# ÍNDICE DE TABLAS

## Índice de Tablas

TABLA 1: DETERMINACIÓN Y DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.....	36
TABLA 2: DESCRIPCIÓN DEL CASO DE USO GESTIONAR EJERCICIOS.....	42
TABLA 3: DESCRIPCIÓN DE LA TABLA USUARIO DE LA BD.....	50
TABLA 4: DESCRIPCIÓN DE LA TABLA SOPA_PALABRA DE LA BD.....	51

# INTRODUCCIÓN

## Introducción

El auge de las Tecnologías de la Información y las Comunicaciones (TIC), ha estado acompañado por un gran avance en la gestión de la información y el conocimiento. Cuba, a la par del creciente papel de las TIC, ha ido desarrollando nuevos proyectos en distintas esferas de la sociedad. El sector de la educación es privilegiado en este sentido, ya que cada día se crean nuevas herramientas educativas con el objetivo de apoyar el proceso de enseñanza-aprendizaje. El software educativo es una de las alternativas que se ha implementado en la medida que avanza la informatización de la sociedad, con el despliegue de recursos informáticos en las escuelas del país, mejorando la calidad de las clases en las aulas y creando el medio adecuado para la profundización del conocimiento.

La Universidad de las Ciencias Informáticas (UCI), creada en el contexto de desarrollo de la Batalla de Ideas, tiene entre sus principales misiones, la formación de un profesional integral y comprometido con la realidad de su entorno. No son pocos los esfuerzos que se ejecutan por parte del colectivo de profesores y estudiantes relacionados con la docencia-producción y su influencia directa en el desarrollo económico del país.

Actualmente, la UCI cuenta con un número considerable de proyectos productivos, nacionales e internacionales, vinculados específicamente con transformaciones en el sector educacional. En la Facultad 4 se encuentra el centro de Tecnologías para la Formación (FORTES), el cual tiene como misión desarrollar tecnologías que permitan ofrecer servicios y productos para la implementación de soluciones de formación, aplicando las TIC a todo tipo de instituciones con diferentes modelos de formación y condiciones tecnológicas. De este modo garantizan la calidad de las soluciones y la formación de los recursos humanos a partir de investigaciones que combinen los elementos pedagógicos y tecnológicos más avanzados, integrando así los procesos de formación, producción e investigación.

Entre los proyectos que la integran, Multisaber-Navegante es uno de los que presenta resultados significativos. El mismo se encuentra inmerso en varias tareas, una de ellas es el desarrollo de una colección de *software* educativo integrada por varios productos, cada uno compuesto por seis módulos fundamentales: Temas, Ejercicios, Juegos, Mediateca, Resultados y Maestro. Estos tienen el propósito de facilitar y complementar el proceso de enseñanza-aprendizaje en los niños de la enseñanza primaria.

## INTRODUCCIÓN

Conformado por las diferentes aplicaciones incluidas dentro del módulo Juegos, se desarrolló la primera versión de una colección de juegos educativos con el objetivo de explotar la creciente motivación que causan en los niños los videojuegos.

Sin embargo, se ha constatado que a pesar de los esfuerzos realizados y los resultados positivos alcanzados en el presente, los juegos han sufrido transformaciones que no han sido incorporadas en esa primera versión. La gestión de los contenidos se realiza directamente accediendo al panel de control del Sistema Gestor de Base de Datos (SGBD), dificultando el acceso a los usuarios y poniendo en riesgo la integridad de los datos. La compatibilidad con navegadores es otro aspecto crucial pues solo puede desplegarse sobre Mozilla Firefox, provocando una innecesaria dependencia hacia software de terceros. La colección está desarrollada para utilizar únicamente a MySQL como SGBD. Por último, el framework utilizado para el desarrollo con JavaScript, Dojotoolkit en su versión 1.1.0, presenta fallos atendiendo a la obsolescencia de dicha versión.

Por lo anteriormente planteado se formula como **problema a resolver**: ¿Cómo lograr una Colección de Juegos Educativos multiplataforma estable y que permita la gestión de los contenidos?

Se plantea como **objeto de estudio** el software educativo. Para dar solución al problema planteado se define como **objetivo general** lograr una Colección de Juegos Educativos multiplataforma, estable y que permita la gestión de los contenidos. Específicamente la investigación centrará su **campo de acción** en una colección multiplataforma de juegos educativos.

Con el desarrollo de esta investigación se espera como **resultado** obtener una Colección de Juegos Educativos con la cual los niños puedan adquirir conocimientos, interactuando y divirtiéndose con la misma. A su vez, se logrará un producto capaz de gestionar los contenidos presentes en esta Colección.

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

- Analizar el estado del arte sobre el desarrollo de juegos educativos, la gestión de contenidos y sus conceptos asociados.
- Realizar el análisis y diseño de la colección multiplataforma de juegos educativos.
- Implementar la colección multiplataforma de juegos educativos.

## INTRODUCCIÓN

Con el propósito de guiar y perfilar el trabajo hacia el alcance de los objetivos trazados, se definieron las siguientes **tareas de investigación**:

- Revisión de la documentación relacionada con el tema a investigar en el mundo, Cuba y la UCI.
- Análisis crítico de las soluciones orientadas a la gestión de contenidos educativos.
- Selección de las herramientas y tecnologías a utilizar.
- Realización del modelo de dominio.
- Definición de los requisitos funcionales y no funcionales.
- Modelación de los diagramas de casos de uso del sistema.
- Descripción de los casos de uso del sistema.
- Modelación de los diagramas de clases del análisis.
- Modelación de los diagramas de clases del diseño.
- Modelación de los diagramas de la fase de implementación.
- Implementación de las funcionalidades identificadas.

Para dar solución a las tareas antes mencionadas se utilizaron los siguientes **métodos de investigación**:

### Métodos teóricos

- **Análítico–Sintético**: se utilizó en el estudio de la base teórica y de las herramientas y tecnologías a emplear, para definir las características esenciales de las mismas y sus relaciones.
- **Análisis Documental**: facilitó el análisis y la sistematización del tema a partir de documentos de autores extranjeros y nacionales.
- **Histórico–Lógico**: para sistematizar las tendencias históricas y actuales en el desarrollo de software educativo en Cuba y el mundo y determinar su influencia en el problema actual de la investigación.
- **Modelación**: para la mejor comprensión del problema se han realizado abstracciones utilizando modelos que tienen una estrecha relación con el objeto que se estudia, estos modelos representan las posibles soluciones que tendrá el problema.

### Métodos empíricos:

- **Entrevista**: permitió la exploración y constatación del estado actual del tema investigado.

# INTRODUCCIÓN

## **Estructura del documento:**

Capítulo 1. Fundamentación teórica. Se fundamentan los elementos teóricos que sustentan la investigación y se realiza un estudio sobre aplicaciones similares existentes. Se hace referencia a las diferentes herramientas y tecnologías utilizadas en el desarrollo de la aplicación, las cuales dan solución al problema planteado.

Capítulo 2. Características del sistema. Se identifican y describen los requisitos funcionales y no funcionales, a la vez que se presentan los actores del sistema y diagramas de casos de uso del mismo, acompañados de la descripción textual de cada uno. Además, se hace un esbozo general de cómo funcionará el sistema a través de un diagrama de dominio.

Capítulo 3. Análisis y diseño. Se describe cómo deberá ser realizada la aplicación partiendo de lo descrito en el capítulo anterior. Además, se presentan los diagramas de clases del análisis y el diseño, que reflejan de una forma más clara cómo va a funcionar el sistema y qué clases estarán presentes en el mismo.

Capítulo 4. Implementación. Se enfoca en la construcción de la solución, explicando los elementos principales del flujo de trabajo de implementación.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

## Capítulo 1. Fundamentación teórica

### **1.1 Introducción**

En este capítulo se abordará la situación actual de los juegos educativos, a fin de determinar las fortalezas y debilidades que presentan. Se realizará el análisis de herramientas para la generación de cuestionarios, lo que constituye un pilar fundamental para la retroalimentación de cómo gestionar las diferentes tipologías de ejercicios. A su vez se realizará un estudio sobre las herramientas y tecnologías más comunes que se utilizan para el desarrollo de aplicaciones educativas, con el objetivo de determinar cuáles son las idóneas para ser utilizadas en la realización del sistema que se propone.

### **1.2 Análisis de soluciones similares existentes**

Teniendo en cuenta los requisitos de la nueva versión de la Colección de Juegos Educativos, se hace necesario un estudio que abarque las soluciones similares de juegos educativos, así como las herramientas para generar cuestionarios. Ambos elementos son de vital importancia a fin de obtener información sobre cómo se desarrollan y qué elementos fundamentales se tienen en cuenta en este tipo de aplicaciones. Para una mejor comprensión se hace el análisis de conceptos asociados.

#### **1.2.1 Juegos educativos**

Es posible referirse a los juegos educativos como un apartado especial dentro del software educativo, por lo tanto, al igual que ellos, estos pueden “tratar las diferentes materias (matemáticas, idiomas, geografía, dibujo,...) de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos,...). Así se logra ofrecer un entorno de trabajo que pueda ser sensible a las circunstancias de los alumnos y relativamente rico en posibilidades de interacción; pero todos comparten cinco características esenciales:

- Son materiales elaborados con una finalidad didáctica, como se deriva de la definición.
- Utilizan el ordenador como soporte en el que los alumnos realizan las actividades que ellos proponen.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Son interactivos, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de información entre el ordenador y los estudiantes.
- Individualizan el trabajo de los estudiantes, ya que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.
- Son fáciles de usar. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un vídeo, es decir, son mínimos; aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer.” [1]

### **JUAdEA:2**

Software educativo titulado “Jugando se aprende de Educación Ambiental”. Su tema central es la problemática ambiental de la provincia de Cienfuegos. Este software está dirigido a estudiantes de noveno grado, por tener conocimientos ambientales previos, aunque puede ser utilizado por otros grados de este nivel de enseñanza. Consta de 8 módulos: Principal, Introducción, Contenido, Biblioteca, Curiosidades, Juegos, Profesor y Créditos.

“Como fase posterior a su diseño fue necesario el ensamblaje del software mediante la herramienta Sprite Kernel, elaborada por el Grupo de Producción de Software de la Universidad de Cienfuegos, todo esto bajo el lenguaje de programación Object Pascal y utilizando el compilador Borland Delphi 7”. [2]

El diseño de los juegos incluye imágenes y se basa en ejercicios de tipo marcar y completar oraciones. El contenido no se puede editar, es obligatorio trabajar con el predefinido. El software fue implementado para entornos de Windows 32 bits y como principal herramienta en el diseño de sus imágenes fue utilizado el software Adobe Photoshop versión 7.0.

### **Colección de juegos educativos de la Universidad de Jaén**

Colección de juegos educativos basados en una arquitectura Cliente-Servidor y una interfaz web de comunicación con el usuario de la misma. Se ha utilizado el lenguaje Java para la implementación de los juegos, específicamente en la generación de Applets, que no es más que una aplicación Java diseñada para su ejecución en un



## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

navegador web. Poseen dos elementos básicos: el entorno de desarrollo y el repositorio de preguntas.

La clasificación en el repositorio se ha hecho en función de las asignaturas o cursos. Así, cada asignatura supone un banco de preguntas diferente. “Dentro de cada subconjunto de preguntas perteneciente a un tipo de juego (por ejemplo: Sopa de Letras), se encuentran diferentes campos para albergar la información de la pregunta en sí, a modo de ejemplo: enunciado, respuesta, opciones” [3]. La distribución de los juegos ha sido a través de paquetes SCORM con árboles de juegos.

### **Hiperentorno de aprendizaje Eureka**

Como parte de la Colección Futuro para el preuniversitario, surge el CD-ROM Eureka, destinado a apoyar la enseñanza de la Matemática en este nivel. El mismo está compuesto por siete módulos para dar apoyo a los diversos momentos del proceso docente-educativo: el módulo Juegos es uno de ellos. Este hiperentorno permite la gestión de usuarios pero no del contenido.

El módulo Juegos permite la interacción de hasta cuatro estudiantes e incluye los siguientes:

**Acróstico:** similar a este popular entretenimiento, donde el estudiante debe ubicar palabras relacionadas con las diferentes temáticas que se abordan en el CD-ROM. Contiene una base de datos de 24 acrósticos con 157 palabras.

**El texto escondido:** entretenimiento en el cual el estudiante debe encontrar una palabra, concepto, etc., en un número determinado de intentos. Contiene una base de datos de 211 términos.

**Encontrando el personaje:** similar al juego “El texto escondido”, donde el estudiante debe encontrar el nombre de un matemático famoso, a partir de pistas que se le ofrecen para llegar a la respuesta correcta. Contiene una base de datos de 138 personajes.

**Sopa de letras:** tal y como el tradicional juego homónimo, brinda la posibilidad de entretenerse encontrando palabras relacionadas con las matemáticas y que se estudian en el programa de preuniversitario. Contiene una base de datos de 35 sopas de letras con 154 términos. [4]

### **GCompris**

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

“GCompris es un programa educativo especialmente diseñado para niños de entre 2 y 10 años. Es software libre y está disponible para Windows, Mac y Linux. Contiene una colección formada por más de 50 actividades y juegos dedicados a tratar todo tipo de temas:

- Uso del ordenador y sus periféricos: acostumbrarse al teclado, ratón.
- Álgebra: memoria, números, tablas, imágenes.
- Ciencia: el ciclo del agua, electricidad, canales, el submarino.
- Geografía: países y mapas.
- Juegos: ajedrez, memoria, fútbol, rompecabezas.
- Lectura: diferentes ejercicios de lectura.
- Otros: las horas, dibujos vectoriales.

Muchas actividades y juegos tienen varios niveles de dificultad, por lo que pueden ser usados por personas de casi cualquier edad. Los niños son usuarios muy especiales y exigentes, y la interfaz está muy trabajada para que sea lo más sencilla, agradable y cómoda posible”. [5]

Cuenta con un panel de administración para la creación de perfiles, grupos y clases. Permite seleccionar qué actividades estarán disponibles para cada perfil. El software está presente en los repositorios de Ubuntu y se incluye por defecto en varias de las distribuciones con fines educativos. Es una aplicación de escritorio.

### **Süchiki Walekerü**

“El software está constituido por cuatro módulos: escuchar el cuento, jugar con el cuento, ayuda y salida del sistema. En el módulo de juegos hay una serie de actividades que en su mayoría buscan incentivar en los niños y maestros el uso de la lectura y escritura del wayuunaiki<sup>1</sup>; tales actividades están conformadas por identificación de vocales, sopa de letras, ejercicios de completar, escribir palabras luego de escuchadas, asociar pares de palabras con imagen, ordenar oraciones para formar párrafos con sentido, juegos de memoria con palabras y rompecabezas, entre otros. El software se concibió y diseñó para hablantes del wayuunaiki y para los que no, siempre y cuando vivan o estén estrechamente interrelacionados con la cultura del pueblo wayuu”. [6]

---

<sup>1</sup> Idioma guajiro empleado en La Guaira, Colombia y en Zulia, Venezuela.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Su objetivo es incentivar la lectura y escritura del wayuunaiki y está constituido por 12 tipos de juegos que en total conforman 23 juegos. El programa seleccionado para desarrollar el software fue el Flash MX de Macromedia.

Partiendo de las características de las colecciones descritas es posible resaltar elementos positivos como la utilización de perfiles que garantizan el acceso de los estudiantes solamente a cierta información predefinida por el profesor, brindando nuevas oportunidades de trabajo diferenciado según las necesidades de cada perfil. La utilización de niveles de dificultad para controlar la complejidad del juego constituye también un aspecto interesante.

Una característica común que arroja la investigación es la tendencia a desarrollar las colecciones de juegos, o en algunos casos el módulo juegos, sin tener presente que se consumen los contenidos predefinidos en la Base de Datos (BD) o repositorio de la aplicación; pasando por alto la importancia de brindarle al profesor el acceso necesario para editar, insertar o eliminar aquellos que considere necesarios de acuerdo a las necesidades específicas de sus estudiantes, limitando la adaptabilidad y funcionalidad del sistema. La selección de las tecnologías es un aspecto a tener en cuenta, puesto que en la confección de algunos de estos juegos se ha optado por utilizar software propietario o por centrarse solo en la compatibilidad con el Sistema Operativo (SO) Microsoft Windows.

### **1.2.2 Herramientas para generar cuestionarios**

#### **Questionmark Perception**

“Es un sistema de administración de evaluaciones completo que permite elaborar preguntas y organizarlas en exámenes, prácticas con preguntas y respuestas, pruebas o encuestas”. [7] Es compatible con los principales estándares, entre ellos: IMS QTI, AICC, SCORM y HR-XML.

Permite aplicar evaluaciones individuales a grupos o subgrupos. Los candidatos y entrevistados pueden iniciar una sesión usando un nombre de usuario y contraseña si así lo desean, para presentar una prueba o responder a una encuesta.

Requisitos del sistema:

- Administrador de autoría de Perception: Windows 2000/XP/2003.
- Servidor Perception: Windows 2000 Server o Windows 2003 Server.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Si se administraran más de 100 evaluaciones diarias, se necesitaría un servidor con una BD aparte que ejecute MS SQL Server u Oracle.

### LimeSurvey

“Es una plataforma web para la administración de encuestas en línea. Tiene establecido roles para gestión, de modo que los usuarios con privilegios de administración tendrán acceso a las diferentes opciones de gestión de la herramienta (según los permisos que se les haya conferido); sin embargo la mayoría de las personas que accederán a la página web únicamente podrán utilizar la herramienta (en este caso completar una encuesta), pero sin posibilidad de alterar aspecto alguno de la misma.” [8]

Entre los diferentes tipos de preguntas disponibles en la plataforma de encuestas se encuentran: Matrices, Opción única, Opción múltiple, Textos y Otros. Cabe destacar que lo anteriormente mencionado son las categorías principales bajo las cuales se agrupan variaciones de ese tipo de pregunta, en el caso de la categoría Otros, se encuentran las opciones de Ordenar Filas y Mensaje o comentario. Es software libre (licencia GNU/GPL v2 o posterior) y es perfectamente posible que todo el sistema necesario para lograr el funcionamiento de estas herramientas sea software libre.

### Hot Potatoes

“Hot Potatoes es un conjunto de seis herramientas de autor desarrollado por el equipo del University of Victoria CALL Laboratory Research and Development, que permiten elaborar ejercicios interactivos basados en páginas Web de seis tipos básicos. La interactividad de los ejercicios se consigue mediante JavaScript.

Descripción de los componentes:

**BC** crea ejercicios de elección múltiple. Cada pregunta puede tener tantas respuestas como se quiera y cualquier número de ellas puede ser correcta. En contestación a cada respuesta se da al estudiante una retroalimentación específica y aparece el porcentaje de aciertos cada vez que se selecciona una respuesta correcta. Tanto en **JBC** como en el resto de los programas es posible incluir una lectura que el alumno efectuará antes de realizar los ejercicios.

**JCloze** genera ejercicios de rellenar huecos. Se puede poner un número ilimitado de posibles respuestas correctas para cada hueco y el estudiante puede pedir

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

ayuda si tiene dudas y se le mostrará una letra de la respuesta correcta cada vez que pulse el botón de ayuda. Una pista específica puede ser también incluida para cada hueco.

El programa permite poner los huecos en palabras seleccionadas por el usuario o hacer el proceso automático de forma que se generen huecos cada “n” palabras de un texto. También se incluye puntuación automática.

**JCross** crea crucigramas, se puede usar una cuadrícula de cualquier tamaño. Como en **JQuiz** y **JCloze**, un botón de ayuda permite al estudiante solicitar una letra en el caso de que la necesite.

**JMatch** crea ejercicios de emparejamiento u ordenación. Una lista de elementos aparecen en la izquierda (estos pueden ser imágenes o texto), con elementos desordenados a la derecha. Esta aplicación puede ser usada para emparejar vocabulario con imágenes o traducciones, o para ordenar sentencias que forman una secuencia o una conversación.

**JMix** crea ejercicios de reconstrucción de frases o párrafos a partir de palabras desordenadas. Es posible especificar tantas respuestas correctas diferentes como se quiera, basadas en palabras y signos de puntuación de la frase base. Se puede incluir un botón que ayude al estudiante con la siguiente palabra o segmento de la frase si lo necesita.” [9]

Dispone de tres tipos de instaladores:

- Hot Potatoes para Windows 98/ME/NT4/2000/XP/Vista.
- Hot Potatoes para usuarios Linux corriendo Wine.
- Java Hot Potatoes para desplegarlo sobre la máquina virtual de Java.

Hot Potatoes es freeware<sup>2</sup> y puede ser utilizado para cualquier propósito o proyecto, pero no es de código abierto.

### EdiLIM

“EdiLIM es un software para crear materiales educativos, especialmente ejercicios aplicables a la docencia y accesibles en forma web.

Permite crear los siguientes tipos de páginas interactivas:

#### Puzzle

---

<sup>2</sup> Hace referencia al software libre.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Resolver un rompecabezas con la posibilidad de ver la imagen con distintos niveles de transparencia. Las piezas se arrastran con el ratón y se ajustan automáticamente.

### **Sopa de letras**

Para buscar palabras mezcladas entre un cuadro de letras de 10 por 10. Puede aparecer la descripción de las palabras a buscar, directamente las palabras o imágenes. Se debe pulsar sobre la primera letra de la palabra y mover el ratón hasta la última.

### **Parejas**

El conocido juego de buscar parejas destapando los cuadros. Actividad con tiempo ajustable.

### **Preguntas**

Hasta tres preguntas para que el usuario introduzca las respuestas.

### **Respuesta múltiple**

Una pregunta y hasta cuatro respuestas posibles. Hay que pulsar con el ratón en la respuesta correcta.

### **Frases**

Ejercicios con frases, para escribir, escuchar, traducir, etc.

Identificar imágenes/sonidos.

Hasta seis imágenes/sonidos con el objetivo de identificarlos, aunque admite más variantes.

### **Arrastrar textos/arrastrar imágenes**

Mover con el ratón las piezas hasta encajarlas correctamente.

Clasificar textos/imágenes.

Llevar las piezas hasta el grupo que corresponda. Sólo permite clasificar en dos grupos.

### **Rayos X**

Muestra dos imágenes superpuestas, la inferior sólo es visible en el espacio definido por un círculo. Se puede arrastrar el círculo sobre la imagen. Los resultados pueden ser sorprendentes. Esta página puede ser descriptiva si no se define una respuesta.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### **Etiquetas**

Mover las etiquetas hasta el lugar correcto. Esta página admite muchas variantes: solamente descriptiva, con sonidos, escribir en las etiquetas, arrastrar, etc.

### **Mover imágenes**

Arrastrar hasta seis imágenes. Permite encajar figuras, completar imágenes, establecer secuencias, etc.

### **Completar**

Llenar huecos mediante la escritura de la respuesta o moviendo cuadros de texto.

### **Palabra secreta**

Descubrir la palabra oculta pulsando sobre las letras, hasta seis posibilidades de error. Admite imágenes.

### **Actividad externa**

Esta página permite incluir actividades ajenas a Lim.

### **Operaciones**

Página para crear operaciones matemáticas: sumas, restas, multiplicaciones y divisiones.

### **Escoger**

Seleccionar entre varias imágenes/textos.

### **Reloj**

Página para crear actividades sobre el manejo del reloj.

### **Enlaces 2**

Para crear índices o sistemas de navegación.

### **Textos**

Página para trabajar con textos, se pueden seleccionar palabras, ordenar frases, arrastrar fragmentos, entre otros.

### **Etiquetas 2**

Situar textos sobre una imagen." [10]

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Requiere tanto para su funcionamiento como para la visualización de las páginas tener instalado Flash.

Si bien las herramientas mencionadas son perfectamente capaces de generar algunas tipologías de ejercicios, presentan limitaciones; no solo en cuanto a la plataforma o la necesidad de adquirir licencias, sino también en la capacidad de adaptarse a los contenidos específicos de algunos de los juegos incluidos en la colección. Por tanto, se hace necesario la implementación de una herramienta que permita la gestión de todos los contenidos concebidos.

### **1.3 Metodologías de desarrollo**

Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a realizar un software. Para la misma se definen tareas que constituyen las actividades elementales en que se dividen los procesos y permiten ser ejecutadas a través de los procedimientos. En este sentido se definen técnicas que se pueden utilizar una o varias veces y herramientas software que automatizan la aplicación. [11]

En estos últimos años se han desarrollado dos corrientes en lo que respecta a los procesos de desarrollo. Por un lado se encuentran las metodologías tradicionales o pesadas entre las que se encuentran el Rational Unified Process (RUP) y Microsoft Solution Framework (MSF), mientras que por el otro están las metodologías ágiles de las cuales se pueden mencionar Extreme Programming (XP), SCRUM y Dynamic System Development Method (DSDM). La diferencia fundamental entre ellas está reflejada en que mientras las tradicionales intentan lograr el objetivo común por medio de orden y documentación, las ágiles tratan de mejorar la calidad del software a través de una comunicación directa e inmediata entre las personas que intervienen en el proceso. [11]

#### **1.3.1 XP**

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Es definida



## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

especialmente como una metodología adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [12]

Esta metodología intenta reducir la complejidad del SW por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. Presenta una debilidad cuando se utiliza en dominios de aplicaciones complejas o situaciones difíciles en la organización: el rol del cliente no refleja los diferentes intereses, habilidades y fuerzas a las que se enfrentan los programadores durante el desarrollo de proyectos. Además, presenta un diseño evolutivo que hace que se le de poca importancia al análisis como fase independiente, puesto que se trabaja exclusivamente en función de las necesidades del momento. [13]

### 1.3.2 RUP

Es una metodología de desarrollo de software que está basada en componentes e interfaces bien definidas. Es un proceso que puede especializarse para una gran variedad de sistemas de software, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Como características presenta la utilización del UML como lenguaje de modelado, unifica los mejores elementos de metodologías anteriores, se encuentra preparado para desarrollar grandes y complejos proyectos y es orientado a objetos.

RUP propone 9 flujos de trabajo que se encuentran insertados en 2 disciplinas: Disciplina de Desarrollo (Ingeniería de Negocios, Requerimientos, Análisis y Diseño, Implementación y Pruebas) y Disciplina de Soporte (Configuración y administración del cambio, Administración del proyecto, Ambiente y distribución).

Su ciclo de vida se caracteriza por ser:

- Dirigido por Casos de Uso: Los casos de uso describen los requisitos funcionales del sistema desde la perspectiva del usuario. Estos no solo inician el proceso de desarrollo sino que también proporcionan un hilo conductor, en el que se verifica, luego de la implementación, que el producto implemente adecuadamente cada Caso de Uso.
- Centrado en la arquitectura: La arquitectura de un sistema permite tener una visión común entre los desarrolladores y los usuarios. Por otra parte, la arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Iterativo e incremental: RUP propone un proceso iterativo e incremental donde el trabajo se divide en partes más pequeñas. Cada una de estas partes se puede ver como una iteración de la cual se obtiene un incremento que produce un crecimiento en el producto. [14]

### 1.3.3 SCRUM

SCRUM es una metodología ágil de desarrollo que surge como modelo para el desarrollo de productos tecnológicos, aunque también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad, situaciones frecuentes en el desarrollo de determinados sistemas de software. Esta metodología requiere trabajo duro puesto que no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. Además, en ocasiones es necesario complementarla con otras metodologías.

Como metodología ágil:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

Además, emplea las siguientes prácticas de la gestión ágil:

- Revisión de las iteraciones
- Desarrollo incremental
- Desarrollo evolutivo
- Auto-organización
- Colaboración

Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. [15]

### 1.3.4 Selección de la metodología de desarrollo

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Se decide utilizar RUP como metodología para que guíe el proceso de desarrollo, por las características que presenta, las facilidades que aporta al proceso y que utiliza conjuntamente el UML para expresar de forma gráfica todos los diagramas de un sistema de software. Además, la selección de esta metodología es un requerimiento necesario por el grupo de desarrollo en la obtención de toda la documentación que genera.

### **1.4 Lenguaje Unificado de Modelado**

Es un lenguaje utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema de software y para especificar modelos no ambiguos y completos. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo. [16]

Está constituido por un conjunto de diagramas, y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que están involucrados en el proceso de desarrollo. Es necesario contar con todos esos diagramas dado que cada uno se dirige a cada tipo de persona implicada en el sistema. [17]

Propiedades que hacen de UML un lenguaje de modelado estándar:

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Ampliamente utilizado por la industria desde su adopción por el Grupo de Gestión de Objetos (OMG, por sus siglas en inglés).
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clases, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboración, que sirven para evaluar el estado de las máquinas. (18)

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### 1.5 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering) son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información. La utilización de estas herramientas afecta a todas las fases del ciclo de vida de un software. Este ciclo de vida se puede considerar como una serie de transformaciones: primero se definen los requisitos del sistema; seguidamente existe un proceso de transformación que hace que la especificación se convierta en un diseño lógico del sistema. Posteriormente, este sufre otro proceso de transformación para lograr un diseño físico, es decir, que responda a la tecnología destino. [19]

#### 1.5.1 Rational Rose

Es una herramienta de modelado visual considerada el producto más completo de la familia *Rational Rose*, que permite crear los diagramas que se van generando durante el proceso de Ingeniería en el Desarrollo de Software.

**Presenta como características adicionales:**

- Generación de código Ada, ANSI C++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo-código configurables.
- Los componentes del modelo se pueden controlar independientemente, lo que permite una gestión y un uso de modelos más granular.
- Funciones de análisis de calidad de código.
- Soporte Enterprise Java Beans™ 2.0.
- Modelado UML para trabajar en diseños de BD, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web. [20]

#### 1.5.2 Visual Paradigm

Visual Paradigm es una herramienta CASE que permite el modelado visual para UML y soporta el ciclo de vida completo del desarrollo de software (análisis y diseño orientados a objetos, construcción, pruebas y despliegue) a través de todo tipo de diagramas.

Como principales características de la herramienta se encuentran:

- Permite dibujar todos los tipos de diagramas de clases.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Genera código inverso y código desde diagramas.
- Genera documentación.
- Agiliza la construcción de aplicaciones con calidad y a un menor coste.
- Posibilita la generación de BD, la transformación de diagramas de Entidad-Relación en tablas de BD, así como ingeniería inversa de BD. [21]

### **1.5.3 Selección de la herramienta de modelado**

Luego del análisis realizado se determina que la herramienta CASE a utilizar en el desarrollo de la Colección de Juegos Educativos es Visual Paradigm, destacando que la misma posee como ventaja que es multiplataforma y a pesar de que no es libre, la UCI posee la licencia para su uso en el desarrollo de aplicaciones. Es importante señalar que esta es la que se utiliza en el proyecto Multisaber-Navegante.

### **1.6 Lenguajes de desarrollo**

Un lenguaje de desarrollo es cualquier lenguaje artificial que se utiliza para definir adecuadamente una secuencia de instrucciones que puedan ser interpretadas y ejecutadas en una computadora. Se asume que las instrucciones así escritas son traducidas luego a un código que la máquina pueda comprender. El proceso de traducción es realizado normalmente por la computadora, usando un programa especializado para tal fin.

Los lenguajes de desarrollo intentan conservar una similitud con el lenguaje humano, con la finalidad de que sean más naturales a quienes los usan. Establecen un conjunto de reglas sintácticas y semánticas, las cuales rigen la estructura del programa de computación que se escribe o edita. De esta forma, permiten a los programadores o desarrolladores, poder especificar de forma precisa los datos sobre los que se van a actuar, su almacenamiento, transmisión y demás acciones a realizar bajo las distintas circunstancias consideradas. [22]

Además, permite al desarrollador comunicarse con el hardware y software existentes mediante el uso de instrucciones, operadores y reglas de sintaxis. Cada lenguaje está desarrollado con un fin específico y en ocasiones estos pueden tener elementos de otros lenguajes.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### 1.6.1 Lenguajes del lado del cliente

#### HTML

HTML, son las siglas de HyperText Markup Language, definiéndolo como un lenguaje de marcado de hipertexto. Brinda etiquetas para incluir elementos diversos, tal es el caso de imágenes y scripts de lenguajes JavaScript y CSS.

Es el lenguaje que se emplea para el desarrollo de páginas de internet y está constituido de elementos que el navegador interpreta y los despliega en la pantalla de acuerdo a su objetivo. Existen elementos para disponer imágenes sobre una página, hipervínculos que permiten dirigirse a otra página, listas, tablas para tabular datos, etc. [23]

#### CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Utilizando CSS se declaran reglas para aplicar estilos visuales a los elementos de una página web y permite separar el contenido de la presentación. [24]

#### JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente es 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 sin necesidad de procesos intermedios. [25]

Cuenta con múltiples aspectos entre los que se pueden mencionar:

- Fácil de aprender.
- Soportado por todos los navegadores.
- Rápido.
- Se ejecuta del lado del cliente. [25]

#### Java

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Es un lenguaje fácil de aprender y orientado a objetos. Posee clases para trabajar con la red y permite abrir, establecer o cerrar conexiones. A diferencia de su antecesor C, se libera del uso de punteros e incluye la recolección de basura para liberar memoria. En el caso de los applets existen fuertes medidas de seguridad que les impiden escribir en la computadora y el acceso a determinados recursos. Sus características más atractivas son la indiferencia a las arquitecturas y la portabilidad de las aplicaciones escritas bajo este lenguaje. Es posible escribir dos tipos de programas: aplicaciones independientes y applets, estos últimos pueden ser incrustados en páginas web, brindando la posibilidad de mostrar animaciones y otras operaciones complejas. [26]

Necesita que la Máquina Virtual de Java esté corriendo en la computadora donde se ejecutan sus aplicaciones.

### 1.6.2 Lenguajes del lado del servidor

#### PHP

PHP (*Hypertext Preprocessor*) es un lenguaje de código abierto muy popular, especialmente adecuado para desarrollo web y que puede ser incrustado en HTML [27]. Puede ser desplegado sobre los principales sistemas operativos (SO), dígame Linux, Microsoft Windows, Mac OS X, RISC OS y variantes de UNIX.

Entre sus características se destacan:

- Soporta la programación orientada a objetos (POO).
- El código es ejecutado en el servidor.
- Fácil de aprender.
- Código abierto.
- Amplia comunidad de desarrollo.
- Soporta la mayoría de los servidores web: Apache, Microsoft IIS, Personal Web Server, iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd.
- Soporta un gran número de BD: Oracle, mSQL, MySQL, ODBC, Sybase, PostgreSQL.

#### Python

Se ejecuta sobre Windows, Linux/Unix, Mac OS X, y también ha sido portado a las máquinas virtuales de Java y .NET. Cuenta con amplia documentación y es fácil de

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

aprender. Liberado bajo licencia de código abierto, por lo que puede ser utilizado con fines comerciales sin problemas.

Se caracteriza por:

- Sintaxis clara.
- Fuertes capacidades de introspección.
- Orientado a objetos.
- Excepciones basadas en el manejo de errores.
- Alto nivel de los tipos de datos dinámicos.
- Extensas librerías estándar y módulos de terceros para prácticamente todas las tareas.
- Extensiones y módulos escritos en C, C++ (o Java para Jython, o .NET para IronPython). [28]

### **1.6.3 Selección de los lenguajes de desarrollo**

Se establece como lenguaje del lado del cliente para el desarrollo de la aplicación a JavaScript frente a Java, teniendo en cuenta la posibilidad de reutilizar código fuente de la Colección de Juegos Educativos y su capacidad de interactuar con los elementos que componen la página web, aspecto imposible si se utilizaran applets. El hecho de que Java depende para su funcionamiento de que la Máquina Virtual esté desplegada en el cliente constituye también un elemento de peso. En cuanto a los lenguajes HTML y CSS, estos son indispensables para desarrollar una aplicación web. Como lenguaje del lado del servidor se selecciona a PHP, partiendo de esta decisión es posible reutilizar parte del código fuente existente, publicar la aplicación sobre una amplia gama de servidores y obtener el producto sin atrasos debido a la necesaria superación del desarrollador. Este lenguaje permite con sus características actuales realizar cada uno de los requisitos expresados por el cliente.

## **1.7 Herramientas de desarrollo**

### **1.7.1 Netbeans IDE**

Es un entorno de desarrollo integrado, liberado bajo licencia de código abierto. Es multiplataforma y permite crear proyectos para la web, el escritorio o para plataformas móviles. Soporta lenguajes como PHP, JavaScript y Ajax, Groovy y Grails, y C++. Extensible mediante plugins permite ajustarlo a las características



## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

propias de cada desarrollador. Cuenta con una amplia comunidad de desarrolladores así como con una amplia documentación.

Esta herramienta presenta otras características:

- Integra control de versiones.
- Permite trazar código PHP.
- Disponible en varios idiomas, el Español entre ellos.
- Soporte para frameworks JavaScript como Dojotoolkit y JQuery. [29]

### **1.7.2 Eclipse IDE**

Permite desarrollar en lenguajes como: Java, C/C++, PHP, JavaScript y Ruby, entre otros. Cuenta con una extensa y prestigiosa comunidad de desarrolladores y se encuentra bajo licencia de código abierto. Ampliamente extensible mediante plugins, es una solución altamente flexible para diversos entornos de desarrollo.

Incorpora elementos como:

- Control de versiones.
- Resaltado de sintaxis.
- Selección de idioma.
- Soporte para frameworks JavaScript como Dojotoolkit y JQuery. [30]

### **1.7.3 Selección de la herramienta de desarrollo**

Partiendo de la selección de herramientas se decide la utilización de NetBeans IDE destacando que presenta características similares al Eclipse IDE, pero además su utilización dentro del proyecto Multisaber-Navegante ha demostrado su eficiencia en el desarrollo de software. El completamiento de código es superior, así como la interfaz.

### **1.8 Framework**

FrameWork es un concepto sumamente genérico, se refiere a ambiente de trabajo y ejecución. En general, los framework son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución). Permiten al desarrollador reutilizar clases y procedimientos, centrándose en las características específicas de lo que desea desarrollar. Agilizan el desarrollo al proveer capas de abstracción que facilitan la

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

codificación de las soluciones. A su vez, agrupan un conjunto de librerías y proveen una estructura para el producto.

### 1.8.1 Framework del lado del cliente

#### Dojotoolkit

Es un completo framework para crear aplicaciones basadas en JavaScript y Ajax. Su licencia es de código abierto. Está compuesto por Widgets que son componentes de código en JavaScript pre-empaquetados que pueden ser utilizados para enriquecer sitios web con varias características que trabajan a través de la mayoría de los navegadores, tales como: Menús, Tabs, Tooltips y Tablas ordenables. [32]

Entre sus características se encuentran las siguientes:

- Realiza consultas al DOM vía selectores de CSS3.
- Brinda utilidades para el trabajo con AJAX.
- Normaliza los eventos sobre varios navegadores.
- Desarrollado, corregido y probado por una amplia comunidad de desarrolladores.
- Permite utilización de otras librerías y frameworks.
- Permite crear animaciones y efectos, generar gráficos, drag and drop (arrastra y suelta), soporta internacionalización.
- Simula clases y herencias.
- Incluye herramientas para la compresión de código.
- Garantiza la portabilidad. [32]

#### JQuery

Es actualmente la librería JavaScript más popular y difundida. Entre sus características se pueden mencionar:

- Acceso a los elementos en el documento. Provee un robusto y eficiente mecanismo de selección para obtener la pieza exacta del documento que se desea inspeccionar o manipular.
- Modificar la apariencia de la página web. Puede cambiar las clases o estilos individuales aplicados a una porción del documento después que la página ha sido cargada.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Organizar el contenido de un documento. El texto puede ser cambiado, imágenes pueden ser insertadas o eliminadas, listas pueden ser reordenadas, o la estructura completa del HTML puede ser reescrita.
- Responder a la interacción del usuario. Ofrece una elegante forma de interceptar gran variedad de eventos, como cuando el usuario da clic sobre un enlace. Al mismo tiempo se remueven las inconsistencias del navegador.
- Animar cambios hechos en el documento. Para implementar efectivamente con los comportamientos interactivos, el diseñador debe proveer una respuesta visual al usuario. La librería JQuery facilita esto brindando un arreglo de efectos como desvanecer y mover.
- Obtener información desde el servidor sin recargar la página. Este patrón de código es conocido como Asynchronous JavaScript And XML (AJAX).
- jQuery elimina la complejidad de un navegar específico para este proceso, permitiendo a los desarrolladores enfocarse en las funcionalidades finales.
- Simplifica tareas comunes con JavaScript. Facilita tareas como la iteración sobre arreglos y su manipulación.
- Es ampliamente extensible mediante plugins y cuenta con una fuerte comunidad. [33]

### 1.8.2 Framework del lado del servidor

#### CodeIgniter

CodeIgniter es un potente framework PHP construido para brindar un conjunto de herramientas que agilicen y optimicen el desarrollo web. Utiliza arquitectura Modelo Vista Controlador (MVC) y permite trabajar con diferentes gestores de BD, cuenta además con una comunidad activa de desarrollo.

Entre sus principales características contempla:

- Tamaño reducido.
- Alto rendimiento.
- Compatibilidad con variedad de versiones y configuraciones de PHP.
- Necesita una configuración mínima.
- No requiere el uso de la línea de comandos.
- No obliga a utilizar un lenguaje de plantillas.
- Provee documentación clara y exhaustiva.
- Extensible mediante librerías, helpers (ayudantes). [34]

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

### **Symfony**

Es un framework PHP de código abierto para el desarrollo de aplicaciones web. Está diseñado para ser compatible con la mayoría de gestores de BD como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Ejecutable tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows.

Sus principales características son:

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del SGBD.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.

[35]

### **1.8.3 Selección de los framework**

Se selecciona como framework del lado del cliente a DojoToolkit debido a su potente manejo de clases que lo hacen perfecto para desarrollar, tiene además características similares a JQuery pero no necesita ser extendido a través de plugins de terceros, cuenta también con herramientas que permiten la ofuscación del código reduciendo considerablemente el tamaño de los scripts. Al utilizar DojoToolkit es posible reutilizar componentes desarrollados en otros productos del proyecto Multisaber-Navegante. Analizando sus características, se decide emplear a CodeIgniter ya que está orientado al desarrollo rápido de aplicaciones web pequeñas o medianas, ha demostrado ser mucho más fácil de utilizar y desplegar, además de requerir únicamente conocimientos sobre el lenguaje PHP.

## **1.9 Gestor de Base de Datos**

### **1.9.1 MySQL**

MySQL es el sistema gestor de BD de código abierto más utilizado del mercado. Este gestor se caracteriza por su facilidad de uso, velocidad y flexibilidad para funcionar en diferentes sistemas operativos.

Permite el acceso simultáneo por varios usuarios a las BD, además gestiona el acceso mediante permisos y privilegios.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Las principales características de este gestor de BD son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords (contraseñas), manteniendo un muy buen nivel de seguridad en los datos.
- Portabilidad: SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando este son fácilmente portables a otros.
- Sistemas y plataformas, que unido al uso de C/C++ proporciona una enorme portabilidad. [36]

### **1.9.2 PostgreSQL**

Es un gestor objeto-relacional, de propósito general, multiusuario y de código abierto, liberado bajo la licencia BSD, que soporta gran parte del estándar SQL y ofrece modernas características como consultas complejas, disparadores, vistas, integridad transaccional, control de concurrencia multiversión y que puede ser extendido por el usuario añadiendo tipos de datos, operadores, funciones de agregados, funciones ventanas o funciones recursivas, métodos de indexado y lenguajes procedurales.

PostgreSQL cuenta con una amplia comunidad en Cuba y el mundo. Es uno de los gestores más potentes y estables. Puede ser desplegado en SO como Microsoft Windows, UNIX y Linux. [37]

### **1.9.3 Selección del Gestor de BD**

Teniendo presente la selección de CodeIgniter como framework es posible desarrollar la aplicación para que interactúe con un SGBD y posteriormente utilizar otro ya sea por requerimiento del cliente o para reutilizar recursos que ya esté ofreciendo el servidor; el único cambio que habría que realizar sería modificar el parámetro que especifica el SGBD a utilizar.

## **CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA**

Por tanto, se elige como SGBD a MySql debido a que es más sencillo de configurar y es posible administrar las BD directamente a través del phpMyAdmin en cualquier navegador.

### **1.10 Servidor Web**

Servidor Web se refiere a un programa que muestra datos a usuarios en forma de páginas web, hipertextos o páginas XHTML mediante la implementación del protocolo HTTP (Hypertext Transfer Protocol). De este modo, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por el cliente HTTP; lo que se conoce como navegador web. El navegador realiza una petición al servidor y este da como respuesta lo que ha solicitado el cliente.

#### **1.10.1 Apache**

Los servidores web Apache a menudo se usan en combinación con el motor de BD MySQL, el lenguaje de scripting PHP, y otros lenguajes de scripting populares como Python y Perl. Esta configuración se denomina LAMP (Linux, Apache, MySQL y Perl/Python/PHP) y conforma una potente y robusta plataforma para el desarrollo y distribución de aplicaciones basadas en la web.

Es actualmente el servidor web más utilizado. Ampliamente extensible a través de módulos y muy configurable. Entre sus principales características destaca la configuración de servidores virtuales que le permiten ejecutar, en la misma máquina, diferentes servidores para diferentes direcciones IP, diferentes nombres de máquina o diferentes puertos. [38]

#### **1.10.2 Internet Information Services**

Este servidor web ofrece un conjunto de servicios para el sistema operativo Microsoft Windows, los cuales proporcionan las herramientas y funciones necesarias para administrar de forma sencilla un servidor web de forma segura. Forma parte de la distribución estándar de Windows, de modo que no se necesita una licencia extra. Este servidor ejecuta scripts "Active Server Pages" (ASP), que es un lenguaje basado en Visual Basic Script. [39]

#### **1.10.3 Selección del servidor web**

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Se elige a Apache como servidor web para el desarrollo de la aplicación atendiendo a la estabilidad y el desempeño que ha mostrado durante el desarrollo de productos similares dentro del proyecto Multisaber-Navegante. No obstante, el producto puede ser desplegado sobre otro servidor si el cliente así lo requiere, ya que tanto los lenguajes seleccionados como los frameworks hacen posible que sea compatible con los principales servidores web existentes.

### **1.11 Conclusiones**

En este capítulo se realizó un estudio detallado de algunas aplicaciones existentes en el mundo y en Cuba, que sirvieron de guía para dar cumplimiento a los objetivos perseguidos en el trabajo de investigación. Se efectuó un análisis de algunas herramientas y tecnologías para la realización de la aplicación, definiéndose RUP como metodología que guíe el proceso de desarrollo, como herramienta para el modelado visual se utilizará Visual Paradigm for UML 8.0 apoyado en el lenguaje UML. Se precisa emplear como lenguajes de desarrollo HTML 5, CSS 3 y JavaScript del lado del cliente y PHP v5.3.5 para el lado del servidor. Se utilizará el framework JavaScript Dojotoolkit v1.7, el framework PHP CodeIgniter v2.0.3 y el Gestor de Bases de Datos MySQL v5.1.16. Por último, como herramienta para el desarrollo de esta aplicación se opta por NetBeans IDE v7.1 y como servidor web Apache v2.2.17.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

### Capítulo 2. Características del Sistema

#### 2.1 Introducción

En este capítulo se aborda el funcionamiento general de la colección mediante un modelo de dominio. Además, se identifican los principales requisitos funcionales y no funcionales que el sistema debe cumplir, se describen los actores que se identificaron al igual que los casos de usos y sus relaciones, mostrados en un diagrama de casos de uso para mejorar su comprensión.

#### 2.2 Modelo de dominio

El modelo de dominio es una forma de representar visualmente clases conceptuales u objetos reales en un dominio de interés. El mismo consiste en un diagrama de clases en el cual a cada objeto se le asigna un nombre (que forma parte de un glosario) con el objetivo de lograr la comprensión y descripción de las clases más importantes dentro del contexto del sistema. [14]

Su principal objetivo es ayudar a los implicados en el desarrollo de un software a utilizar un vocabulario común que posibilite una mejor comprensión entre ellos, para poder entender y describir las clases más importantes dentro del contexto donde se ubica el sistema. Además permite y facilita el levantamiento de requisitos así como definir los procesos y roles más significativos.

##### 2.2.1 Descripción de los conceptos

A continuación se identifican y describen los principales conceptos relacionados con el contexto del problema, los cuales serán utilizados en la confección del modelo de dominio.

**Juegos:** Actividades que tienen como objetivo profundizar en algún contenido en específico, a la par de la diversión y el entretenimiento.

**Ejercicios:** Actividades en específico que permiten ejercitar y comprobar habilidades adquiridas.

**Contenidos:** Agrupa los elementos (imágenes, temas, etc.) que forman parte de los juegos que no contienen ejercicios, como son: Sopa de letras, Rompecabezas, Crucigramas, Tangram, ente otros.

**Jugador:** Persona que interactúa con la colección de juegos educativos. La aplicación permitirá hasta 4 jugadores al mismo tiempo.



## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

**Administrador:** Persona que gestiona los ejercicios y contenidos que serán mostrados a los usuarios en los diferentes juegos.

### 2.2.2 Diagrama de dominio

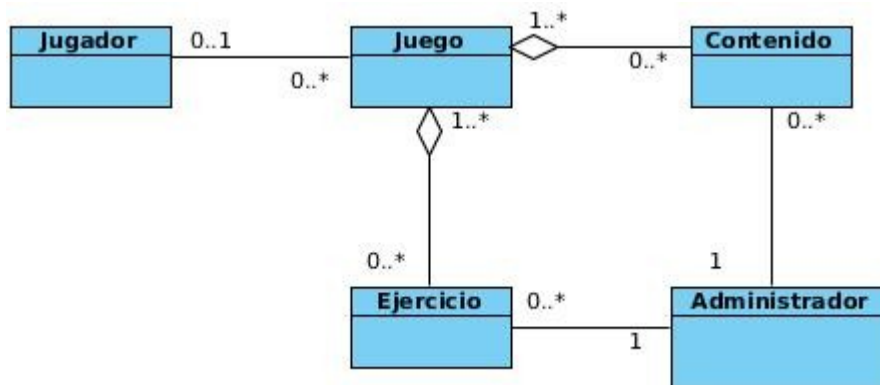


Figura 1: Modelo del Dominio

## 2.3 Especificación de requerimientos del sistema

Los requerimientos son propiedades o restricciones determinadas de forma precisa que deben satisfacerse. [42] Describen los servicios que ha de ofrecer el mismo y las restricciones asociadas a su funcionamiento. Los requisitos se definen en lenguaje natural, se expresan de forma individual y generalmente se enumeran. A su vez, deben ser concisos, consistentes, claros y concretos para evitar ambigüedades.

### 2.3.1 Requerimientos funcionales

Los requerimientos funcionales expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuál va a ser su estado y funcionamiento). [42]

#### RF 1: Gestionar ejercicios

RF 1.1: Adicionar ejercicio

RF 1.2: Modificar ejercicio

RF 1.3: Listar ejercicios

RF 1.4: Eliminar ejercicio

RF 1.5: Buscar ejercicio

RF 1.6: Vista previa

#### RF 2: Gestionar temas

RF 2.1: Adicionar tema

## **CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA**

- RF 2.2:** Modificar tema
- RF 2.3:** Listar temas
- RF 2.4:** Eliminar tema
- RF 2.5:** Buscar tema
- RF 3:** Gestionar perfiles
  - RF 3.1:** Adicionar perfil
  - RF 3.2:** Modificar perfil
  - RF 3.3:** Listar perfiles
  - RF 3.4:** Eliminar perfil
  - RF 3.5:** Buscar perfil
- RF 4:** Gestionar crucigramas
  - RF 4.1:** Adicionar crucigrama
  - RF 4.2:** Modificar crucigrama
  - RF 4.3:** Listar crucigramas
  - RF 4.4:** Eliminar crucigrama
  - RF 4.5:** Buscar crucigrama
  - RF 4.6:** Vista previa
- RF 5:** Gestionar palabras
  - RF 5.1:** Adicionar palabra
  - RF 5.2:** Modificar palabra
  - RF 5.3:** Listar palabras
  - RF 5.4:** Eliminar palabra
  - RF 5.5:** Buscar palabra
- RF 6:** Gestionar usuarios
  - RF 6.1:** Adicionar usuario
  - RF 6.2:** Modificar nombre de usuario
  - RF 6.3:** Modificar clave
  - RF 6.4:** Listar usuario
  - RF 6.5:** Eliminar usuario
  - RF 6.6:** Buscar usuario
- RF 7:** Autenticar jugador
  - RF 7.1:** Seleccionar la cantidad de jugadores
  - RF 7.2:** Mostrar todos los avatares
  - RF 7.3:** Permitir seleccionar avatar
  - RF 7.4:** Mostrar los perfiles

## **CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA**

**RF 7.5:** Permitir seleccionar perfil

**RF 8: Autenticar usuario**

**RF 8.1:** Permitir introducir nombre de usuario

**RF 8.2:** Permitir introducir clave de usuario

**RF 9: Realizar juegos con ejercicios**

**RF 9.1:** Mostrar preguntas

**RF 9.2:** Permitir seleccionar respuestas

**RF 9.3:** Evaluar respuestas

**RF 10: Permitir ver resultados de los juegos**

**RF 10.1:** Mostrar nombre del jugador

**RF 10.2:** Mostrar tiempo

**RF 10.3:** Mostrar efectividad

**RF 10.4:** Mostrar podio

**RF 11: Controlar funcionalidades del juego Ensartando Aros**

**RF 11.1:** Permitir ensartar los aros correctamente

**RF 11.2:** Permitir separar los aros del conjunto

**RF 11.3:** Permitir mover los aros hacia afuera

**RF 11.4:** Permitir balancear aros

**RF 12: Controlar funcionalidades del juego Sopa de Letras**

**RF 12.1:** Seleccionar el o los temas que se desean abordar

**RF 12.2:** Seleccionar la o las sopas de letras que se desean realizar

**RF 12.3:** Seleccionar el nivel de dificultad

**RF 12.4:** Permitir ver las preguntas

**RF 12.5:** Seleccionar palabras de respuesta

**RF 12.6:** Permitir revisar las respuestas

**RF 13: Controlar funcionalidades del juego Crucigramas**

**RF 13.1:** Seleccionar el o los temas que se desean abordar

**RF 13.2:** Seleccionar el o los Crucigramas que se desean realizar

**RF 13.3:** Mostrar preguntas

**RF 13.4:** Permitir introducir respuestas

**RF 13.5:** Permitir revisar la respuesta

**RF 13.6:** Permitir ver la solución

**RF 14: Controlar funcionalidades del juego Rompecabezas**

**RF 14.1:** Seleccionar el o los Rompecabezas que se desean armar

**RF 14.2:** Seleccionar el número de piezas

## **CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA**

**RF 14.3:** Seleccionar el tipo de corte

**RF 14.4:** Permitir rotar una pieza

**RF 14.5:** Mover piezas

**RF 14.6:** Mostrar opciones de ayuda

**RF 14.7:** Mostrar imagen completa

**RF 14.8:** Cargar las piezas de un Rompecabezas

### **RF 15: Controlar funcionalidades del juego Ludo**

**RF 15.1:** Permitir mover los dados

**RF 15.2:** Permitir mover la (s) ficha (s)

**RF 15.3:** Mostrar imagen del juego

### **RF 16: Controlar funcionalidades del juego Tangram**

**RF 16.1:** Seleccionar la cantidad de Tangram que se desean armar y colorear

**RF 16.2:** Mostrar imagen del Tangram

**RF 16.3:** Mostrar colores

**RF 16.4:** Permitir seleccionar colores

**RF 16.5:** Permitir mover piezas

**RF 16.6:** Permitir rotar piezas en ambos sentidos

**RF 16.7:** Permitir dar color a las piezas

**RF 16.8:** Permitir revisar respuesta

### **RF 17: Mostrar pantalla principal de los juegos**

**RF 17.1:** Mostrar una imagen de identificación de cada juego

**RF 17.2:** Permitir seleccionar un juego

**RF 17.3:** Permitir la salida del sistema

### **RF 18: Mostrar información del juego**

**RF 18.1:** Mostrar ubicación

**RF 18.2:** Mostrar efectividad

**RF 18.3:** Mostrar tooltips con el tiempo

**RF 18.4:** Mostrar nombre del jugador en turno

**RF 18.5:** Mostrar avatar de cada jugador implicado en el juego

**RF 18.6:** Mostrar avatar del jugador en turno resaltado

**RF 18.7:** Mostrar tooltip con el nombre del jugador en el avatar

**RF 18.8:** Permitir avanzar al siguiente juego

**RF 18.9:** Permitir ir a la pantalla de los resultados

**RF 18.10:** Permitir regresar a la pantalla de los juegos

**RF 19:** Gestionar imagen

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

**RF 19.1:** Subir imagen a un ejercicio

**RF 19.2:** Modificar imagen de un ejercicio

**RF 19.3:** Eliminar imagen de un ejercicio

**RF 20:** Mostrar panel de administración

### 2.3.2 Requerimientos no funcionales

Los requerimientos no funcionales representan las restricciones sobre el espacio de posibles soluciones. Se deben especificar cuantitativamente siempre que se pueda, con el objetivo de poder verificar su cumplimiento.

**RNF1:** Requisitos de hardware

- Procesador Pentium 233 MHz (recomendado 500 MHz o mayor).
- 512 MB de RAM (recomendado 1 GB de RAM o mayor).
- 700 MB de espacio en disco duro.
- Soporte de video que admita resolución de al menos 800x600 y 24 bits.
- Dispositivo de red de al menos 10 Mbits.

**RNF 2:** Requisitos de software

- La Colección será multiplataforma, por lo que podrá ser ejecutada sobre cualquier sistema operativo.
- La aplicación será compatible con los navegadores: Mozilla Firefox, Opera, Google Chrome e Internet Explorer.

**RNF 3:** Requisitos de restricciones en el diseño y la implementación

- Framework del lado del cliente: Dojotoolkit
- Framework del lado del servidor: CodeIgniter
- Lenguaje de desarrollo del lado del cliente: JavaScript
- Lenguaje de desarrollo del lado del servidor: PHP
- Servidor web: Apache
- IDE de desarrollo: Netbeans
- Servidor de BD: MySQL

**RNF 4:** Requisitos de apariencia e interfaz externa

- Los juegos tendrán una interfaz sencilla y amigable.
- La información de los juegos será presentada de forma clara y organizada, permitiendo una correcta interpretación en los usuarios.
- Los juegos se ejecutarán de manera rápida, facilitando una mejor interacción entre usuario y juego.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

- El diseño de la interfaz gráfica garantizará la distinción visual de los principales elementos de los juegos.
- Cada juego tendrá una imagen que lo identifique.

### **RNF 5:** Requisitos legales

- Las medias que se utilicen en el producto deben tener el permiso legal de sus autores y su aprobación para hacer uso de ellas.

### **RNF 6:** Requisitos de seguridad

- El sistema deberá tener el control de la información a la que accede cada usuario.
- No se denegará el acceso a la aplicación a ningún usuario.

### **RNF 7:** Requisitos de usabilidad

- El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre computación.

### **RNF 8:** Requisitos de soporte

- Se realizará transferencia tecnológica de la colección a los clientes.

## **2.4 Modelo del sistema**

### **2.4.1 Definición de actores del sistema**

Un actor del sistema es un rol que desempeña un usuario o sistema externo al interactuar con el sistema que se modela, por tanto el actor representa terceros fuera del sistema, que interactúan con él. A continuación se hace una descripción de los actores que interactúan con el sistema:

Actores	Descripción
Jugador	Persona que inicializa la interacción con la colección de juegos educativos.
Administrador	Persona que gestiona los contenidos.

*Tabla 1: Determinación y descripción de los actores del sistema.*

### **2.4.2 Diagrama de casos de uso del sistema**

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto. Un diagrama de casos de uso explica gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre estos y los casos de

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

uso. Tiene por objeto ofrecer una clase de diagrama contextual que nos permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan. [14]

El sistema constará con dos aplicaciones: una aplicación frontend y una backend. El frontend es la parte pública del sistema, es lo que los usuarios van a ver cuando accedan al mismo; es el encargado de presentar los juegos al jugador. El *backend* es la parte privada de la aplicación; a esta zona solo tendrán acceso los administradores, en ella se gestionarán los contenidos.

A continuación se presentan los diagramas de casos de uso del sistema, los cuales se encuentran agrupados según la organización que presentará la aplicación.

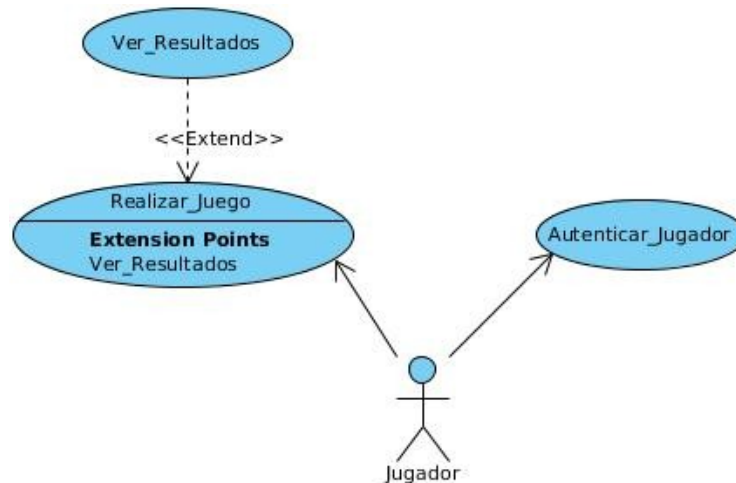


Figura 2: Diagrama de Casos de Uso (Frontend)

Por la complejidad que presenta el diagrama de casos de uso del *backend*, se decidió dividirlo en 3 diagramas para lograr una mejor comprensión del mismo.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

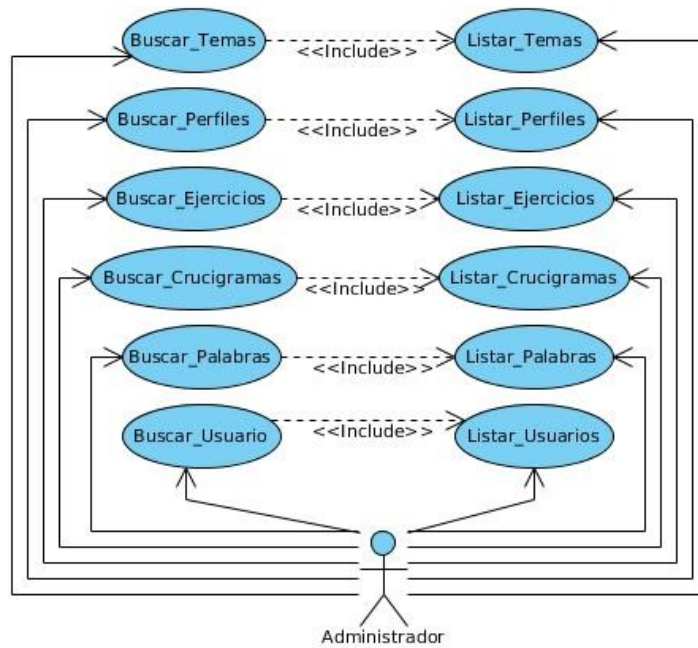


Figura 3: Diagrama de Casos de Uso (Backend1)

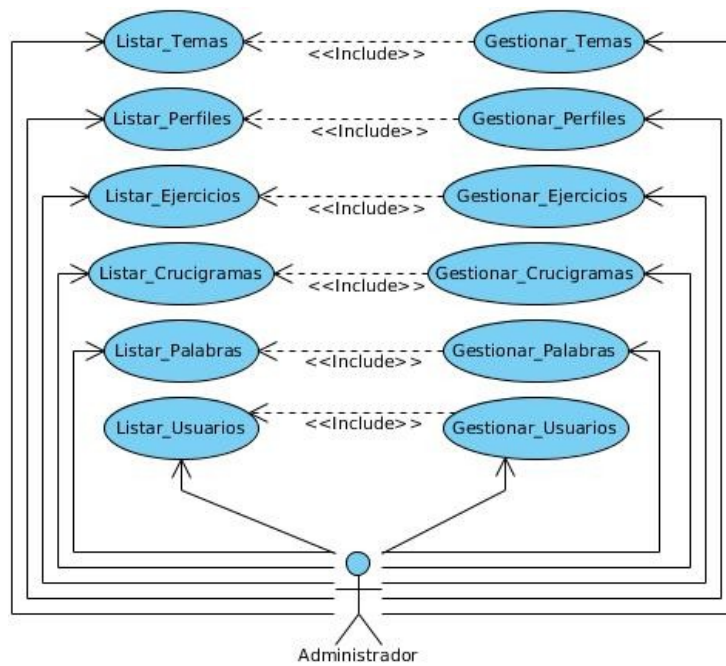


Figura 4: Diagrama de Casos de Uso (Backend2)



## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

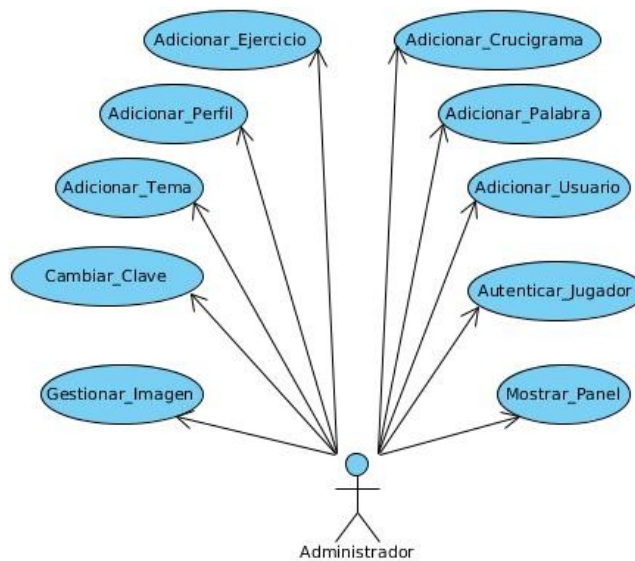


Figura 5: Diagrama de Casos de Uso (Backend3)

### 2.4.3 Descripción de los casos de uso del sistema

<b>Caso de Uso</b>	Gestionar_Ejercicios
<b>Actor</b>	Administrador
<b>Resumen</b>	El caso de uso se inicia cuando el administrador selecciona del listado de ejercicios existente en la BD, la acción de modificar o eliminar algún ejercicio, luego de realizada esta operación finaliza el caso de uso.
<b>Precondiciones</b>	El administrador debe estar autenticado en el sistema. El administrador debe haber listado los ejercicios de la BD.
<b>Referencia</b>	RF 1.2, RF 1.4, RF 1.6
<b>Prioridad</b>	Secundario
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.- El caso de uso se inicia cuando el usuario desea modificar o eliminar un ejercicio.	2.- Permite <ul style="list-style-type: none"> <li>• Modificar un ejercicio. Ver <i>Sección 1 "Modificar ejercicio"</i>.</li> <li>• Eliminar un ejercicio. Ver <i>Sección 2 "Eliminar ejercicio"</i>.</li> <li>• Vista previa de un ejercicio. Ver <i>Sección 3 "Vista previa"</i>.</li> </ul>

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

	3.- El caso de uso termina.
<b>Sección 1 “Modificar ejercicio”: Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1.- Selecciona la opción que permite modificar los datos de un ejercicio.</p> <p>3.- Modifica los datos que desea.</p> <p>4.- Selecciona la opción Aplicar los cambios.</p> <p>5.- Selecciona la opción Guardar los cambios.</p> <p>9.- Selecciona la opción Aceptar.</p>	<p>2.- Muestra los datos del ejercicio y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes:</p> <ul style="list-style-type: none"> <li>• Tipología: Selección Simple, Selección Múltiple o Verdadero o Falso.</li> <li>• Tema</li> <li>• Perfil: Avanzado o Medio</li> <li>• Imagen</li> <li>• Enunciado</li> <li>• Adicionar y/o eliminar opciones de respuesta.</li> <li>• Seleccionar la o las respuestas correctas.</li> </ul> <p>Permite:</p> <ul style="list-style-type: none"> <li>• Aplicar los cambios</li> <li>• Guardar los cambios</li> </ul> <p>6.- Valida que no hayan quedado campos vacíos.</p> <p>7.- Actualiza los datos del ejercicio.</p> <p>8.- Muestra el mensaje de información “Ejercicio modificado correctamente”.</p> <p>Permite seleccionar la opción Aceptar.</p> <p>10.- Finaliza el caso de uso.</p>
<b>Flujo Alternativo</b>	
<b>3. a Carga un archivo que no es una imagen.</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

<p>3. a.2 Selecciona la opción Aceptar.</p>	<p>3. a.1 Muestra el mensaje de información: “El tipo de archivo que está tratando de subir no está permitido”.</p> <p>Permite seleccionar la opción Aceptar.</p> <p>3. a.3 Regresa al paso 2 del Flujo Básico.</p>
<p><b>3. b Carga una imagen que excede el tamaño establecido.</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del sistema</b></p>
<p>3. b.2 Selecciona la opción Aceptar.</p>	<p>3. b.1 Muestra el mensaje de información: “El archivo que está tratando de subir excede el tamaño permitido”.</p> <p>Permite seleccionar la opción Aceptar.</p> <p>3. b.3 Regresa al paso 2 del Flujo Básico.</p>
<p><b>4. a No existen opciones de respuesta.</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del sistema</b></p>
<p>4. a.2 Selecciona la opción Aceptar.</p>	<p>4. a.1 Muestra el mensaje de información: “Necesita adicionar al menos una opción”.</p> <p>Permite seleccionar la opción Aceptar.</p> <p>4. a.3 Regresa al paso 3 del Flujo Básico.</p>
<p><b>6. a No se definió enunciado para el ejercicio.</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>6. a.2 Seleccionar la opción Aceptar.</p>	<p>6. a.1 Muestra el mensaje de información: “Enunciado no establecido”.</p> <p>Permite seleccionar la opción Aceptar.</p> <p>6. a.3 Regresa al paso 3 del Flujo Básico.</p>
<p><b>Sección 2 “Eliminar ejercicio”: Flujo Normal de Eventos</b></p>	
<p><b>Acción del Actor</b></p>	<p><b>Respuesta del Sistema</b></p>
<p>1.- Selecciona la opción que permite eliminar un ejercicio.</p>	<p>2.- Muestra el mensaje de confirmación: “¿Está seguro que desea eliminar el ejercicio?”.</p> <p>Permite:</p>

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

<p>3.- Selecciona la opción Aceptar.</p> <p>7.- Selecciona la opción Aceptar.</p>	<ul style="list-style-type: none"> <li>• Aceptar</li> <li>• Cancelar</li> </ul> <p>4.- Elimina el ejercicio.</p> <p>5.- Actualiza el listado de ejercicios.</p> <p>6.- Muestra el mensaje de información: “Eliminado correctamente”.</p> <p>Permite seleccionar la opción Aceptar.</p> <p>8.- Finaliza el caso de uso.</p>
<b>Flujo Alterno</b>	
<b>4. a Selecciona la opción Cancelar.</b>	
<b>Acción del Actor</b>	<b>Respuesta del sistema</b>
	4. a.1 Finaliza el caso de uso.

Tabla 2: Descripción del caso de uso Gestionar ejercicios.

### 2.5 Conclusiones

En este capítulo se mostraron las principales clases del dominio y se definieron los actores que interactúan con el sistema, además de analizar las características y funciones fundamentales del sistema para el desarrollo de la Colección de Juegos Educativos. Estos se representaron mediante un diagrama de casos de uso, llegando así a las descripciones detalladas de los mismos. Una vez realizado esto es posible comenzar a realizar el análisis y diseño de la aplicación teniendo en cuenta los requerimientos especificados en el capítulo.

## CAPÍTULO 3. ANÁLISIS Y DISEÑO

### Capítulo 3. Análisis y Diseño

#### 3.1 Introducción

El presente capítulo está centrado en el flujo de trabajo Análisis y Diseño, representando mediante un grupo de artefactos la descripción del sistema que sirve de entrada para la implementación del mismo. Se realizan los diagramas de clases del análisis y del diseño, se define la estructura de la BD necesaria para el almacenamiento de la información, así como la descripción de cada una de sus tablas.

#### 3.2 ¿Qué es el análisis?

El análisis es uno de los flujos de trabajos desarrollado en la fase de elaboración. Consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que solo se interesa por los requisitos funcionales.

#### 3.3 Modelo de análisis

El modelo de análisis es una especificación detallada de los requisitos y funciona como primera aproximación del modelo de diseño, aunque es un modelo con entidad propia. Se utiliza para crear un sistema robusto y flexible que emplea la reutilización de componentes de manera considerable. [14]

El modelo de análisis debe lograr tres objetivos primarios:

1. Describir lo que requiere el cliente.
2. Establecer una base para la creación de un diseño de software.
3. Definir un conjunto de requisitos que se pueda validar una vez que se construye el software. [14]

##### 3.3.1 Diagramas de clases del análisis

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.

A continuación, se presentan los diagramas de clases del análisis correspondiente a los casos de uso del sistema descritos anteriormente.

## CAPÍTULO 3. ANÁLISIS Y DISEÑO

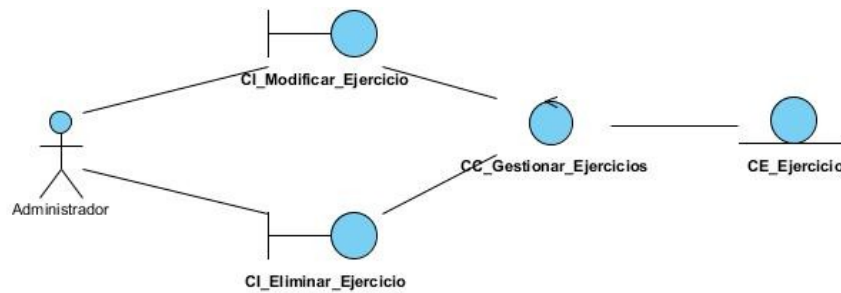


Figura 6: Diagrama de clases del análisis CU Gestionar\_Ejercicios

### 3.3.2 Diagramas de clases de colaboración

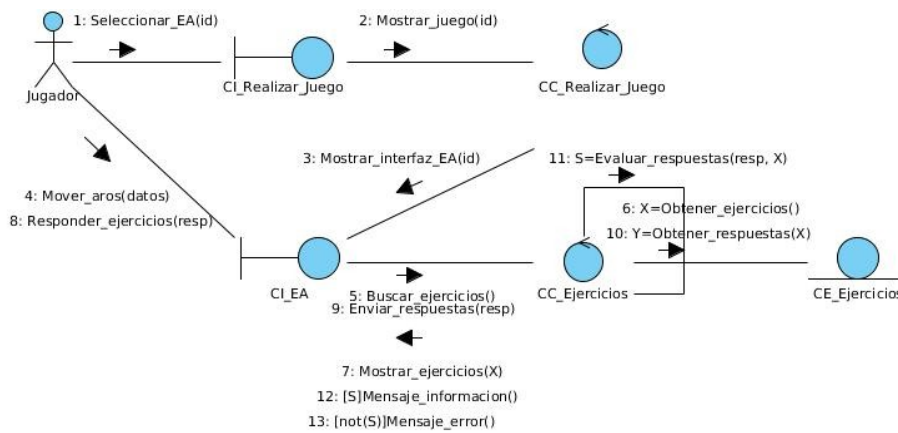


Figura 7: Diagrama de clases de colaboración CU Realizar\_Juego (Escenario Juego\_EA)

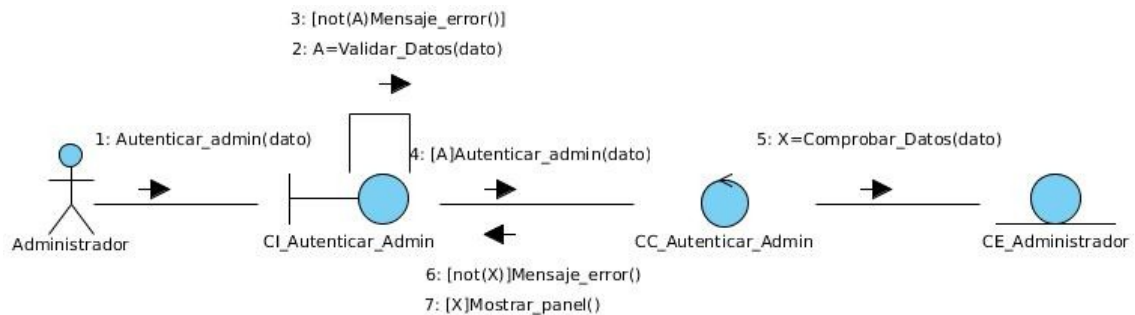


Figura 8: Diagrama de clases de colaboración CU Autenticar\_Admin

### 3.4 Patrón arquitectónico

Los patrones arquitectónicos son patrones del software que se encargan de definir la estructura de un sistema, estos a su vez se componen de subsistemas con sus responsabilidades. Tienen una serie de directivas para organizar los componentes del

## CAPÍTULO 3. ANÁLISIS Y DISEÑO

mismo sistema, con el objetivo de facilitar la tarea del diseño de este. Un patrón arquitectónico se enfoca en dar solución a un problema en específico, de un atributo de calidad, y abarca solo parte de la arquitectura.

### 3.4.1 Modelo Vista Controlador (MVC)

MVC es un enfoque de software que separa la lógica de la aplicación de la presentación. En la práctica, le permite a sus páginas web contener mínimo código ya que la presentación está separada del código PHP.

- El Modelo representa sus estructuras de datos. Típicamente sus clases del modelo contienen funciones que ayudan a devolver, insertar y actualizar información de su BD.
- La Vista es la información que se presenta al usuario. Una vista será normalmente una página web, pero en CodeIgniter, una vista también puede ser un fragmento de página como el encabezado o pie de página. También puede ser una página RSS, o cualquier otro tipo de "página".
- El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web. [34]

### 3.5 Patrones de diseño

Cada patrón describe un problema que ocurre una y otra vez en diferentes entornos, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma. El patrón es un esquema de solución que se aplica a un tipo de problema, esta aplicación del patrón no es mecánica, sino que requiere de adaptación y matices. [42]

Para el desarrollo de la aplicación se utilizaron varios patrones de diseño, algunos de los cuales vienen integrados con el framework utilizado: CodeIgniter. Estos patrones juegan un papel fundamental en la implementación del sistema puesto que proporcionan beneficios a partir del cumplimiento de los objetivos con los que fueron utilizados.

#### 3.5.1 Patrón Singleton (Solitario)

Dentro de la clasificación GoF, es un patrón creacional que garantiza que una clase sólo tenga una única instancia, proporcionando un punto de acceso global a la misma.

## CAPÍTULO 3. ANÁLISIS Y DISEÑO

Algunas ventajas de este patrón:

- Acceso controlado a la única instancia.
- Espacio de nombres reducido.
- Permite el refinamiento de operaciones y la presentación.
- Permite un número variable de instancias.
- Es más flexible que las operaciones de clases. [40]

### 3.5.2 Patrón Facade (Fachada)

El patrón Fachada hace dos cosas: simplifica la interfaz de una clase, y separa esa clase desde el código del cliente que lo utiliza. Las fachadas son el principio básico detrás de casi todas las bibliotecas de JavaScript. Este patrón puede hacer la biblioteca más fácil de entender mediante la creación de métodos de conveniencia, utilidades que permiten a los complejos sistemas ser utilizado de manera más fácil y sencilla. El patrón de la fachada simplifica las tareas comunes o repetitivas, tales como el registro de errores de logueo y realiza un seguimiento de estadísticas de visitas de la página.

Este patrón sirve para simplificar las interfaces complejas, pueden hacer la comprobación de errores detrás de las escenas, limpiar objetos de gran tamaño que ya no están en uso, y presentar las características de un objeto en una forma más fácil de usar. Las fachadas no son estrictamente necesarias: las mismas tareas se pueden realizar sin ellas. Eso es posible puesto que este es un patrón de organización, el cual permite modificar las interfaces de las clases y los objetos para que sean más convenientes. [41]

### 3.5.3 Patrón Observer (Observador)

El objetivo de este patrón es desacoplar la información del objeto (atributos) del comportamiento (métodos), esto se logra creando un objeto que únicamente contiene información básica, y esta delega el comportamiento o las acciones a tomar a un objeto observador, el cual observa el objeto en cuestión, detecta o es notificado por los eventos que suceden en el mismo, y reacciona de una manera específica.

Beneficios que presenta:

- Desacopla los objetos y su comportamiento.
- Reduce la dependencia entre los objetos.
- Permite dinamismo en la creación de observadores. [40]



## CAPÍTULO 3. ANÁLISIS Y DISEÑO

### **3.6 Modelo de diseño**

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a considerar. Este se crea tomando el modelo de análisis como entrada principal, pero se adapta al entorno de implementación elegido. También se debe adaptar para reutilizar sistemas heredados u otros marcos de trabajo desarrollados para el proyecto. [14]

#### **3.6.1 Diagramas de clases del diseño**

Los diagramas de clases muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. El diagrama de clases del diseño es un modelo de objetos que describe la realización de los casos de uso, centrándose en los requisitos funcionales y no funcionales. Para la modelación de la aplicación se proponen como estereotipos del diseño los estándares para modelar aplicaciones web, ya que estas se diferencian un poco del resto de las aplicaciones, representando así las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. [14]

A continuación se presentan solamente dos de los diagramas de clases del diseño de la aplicación puesto que los demás tienen un comportamiento similar en funcionamiento, según las características esenciales de cada uno.

## CAPÍTULO 3. ANÁLISIS Y DISEÑO

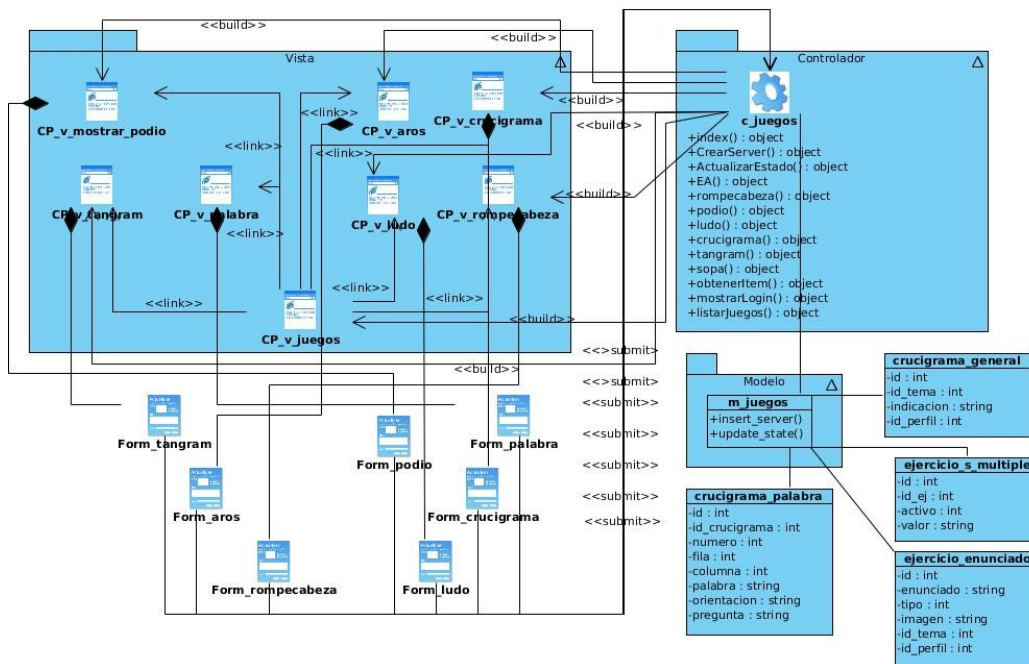


Figura 9: Diagrama de clases del diseño CU Realizar\_Juego

### 3.7 Diseño de la BD

#### 3.7.1 Diagrama de clases persistentes

Una clase persistente se define por contener objetos de característica trascendente en los estados de tiempo y espacio en que se encuentra, ya que existirá durante la ejecución de un programa, manteniendo su información al existir una posible eliminación o colapso del mismo. A su vez, estas clases no necesitan implementar ninguna interfaz ni heredar de otra clase, aunque no restringe realizar el procedimiento anterior. A continuación se muestra el diagrama de clases persistentes de la aplicación:

## CAPÍTULO 3. ANÁLISIS Y DISEÑO

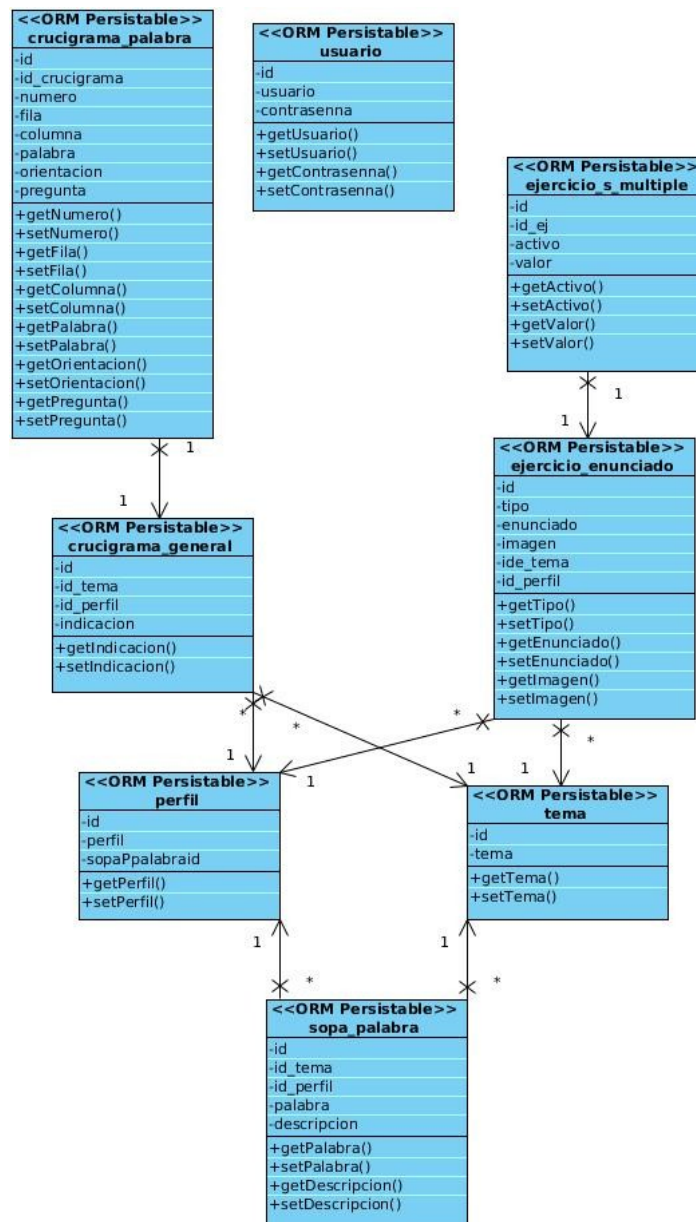


Figura 10: Diagrama de clases persistentes

### 3.7.2 Diagrama Entidad-Relación (ER) de la BD

Un diagrama ER representa el modelado de datos de un sistema de información, basándose en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos objetos. A continuación se muestra el diagrama ER de la presente investigación:

## CAPÍTULO 3. ANÁLISIS Y DISEÑO

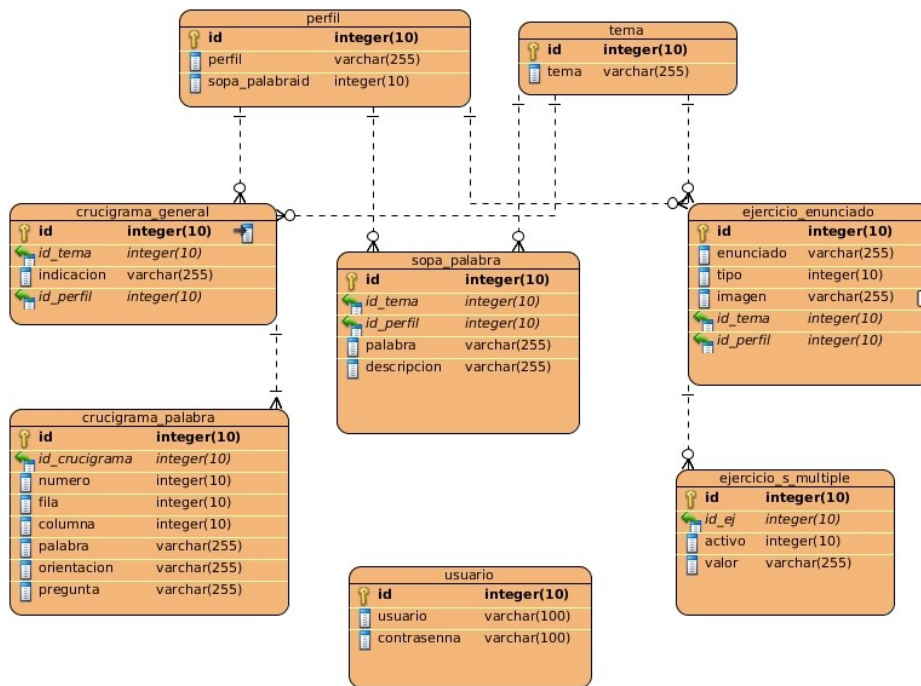


Figura 11: Diagrama Entidad-Relación

### 3.7.3 Descripción de las tablas de la BD

A continuación se presentan las descripciones de las tablas de la BD, con el objetivo de lograr una mejor comprensión de estas y de cada uno de los atributos que la componen.

<b>Nombre:</b>	usuario	
<b>Descripción:</b>	Contiene a los usuarios del sistema con su respectiva clave.	
<b>Atributo:</b>	<b>Tipo:</b>	<b>Descripción:</b>
<ul style="list-style-type: none"> <li>id</li> <li>usuario</li> <li>contrasenna</li> </ul>	<ul style="list-style-type: none"> <li>Integer</li> <li>Varchar</li> <li>Varchar</li> </ul>	<ul style="list-style-type: none"> <li>Identificador del usuario</li> <li>Texto del usuario</li> <li>Clave correspondiente al usuario</li> </ul>

Tabla 3: Descripción de la tabla usuario de la BD

<b>Nombre:</b>	sopa_palabra	
<b>Descripción:</b>	Contiene las palabras con su correspondiente descripción, tema y perfil. Estos datos son consumidos por el juego Sopa de letras.	
<b>Atributo:</b>	<b>Tipo:</b>	<b>Descripción:</b>
<ul style="list-style-type: none"> <li>id</li> <li>id_tema</li> <li>id_perfil</li> </ul>	<ul style="list-style-type: none"> <li>Integer</li> <li>Integer</li> <li>Integer</li> </ul>	<ul style="list-style-type: none"> <li>Identificador de la palabra</li> <li>Identificador del tema al que pertenece la palabra</li> <li>Identificador del perfil al que pertenece la palabra</li> </ul>

## CAPÍTULO 3. ANÁLISIS Y DISEÑO

<ul style="list-style-type: none"><li>• palabra</li><li>• descripcion</li></ul>	<ul style="list-style-type: none"><li>• Varchar</li><li>• Varchar</li></ul>	<ul style="list-style-type: none"><li>• Texto de la palabra</li><li>• Descripción de la palabra</li></ul>
---	---	---

Tabla 4: Descripción de la tabla sopa\_palabra de la BD

### 3.8 Conclusiones

En este capítulo se abordó el análisis y el diseño del sistema para el desarrollo de la Colección de Juegos Educativos. Se desarrollaron los diagramas de clases del análisis y el modelo de diseño teniendo en cuenta el patrón de arquitectura Modelo Vista Controlador. Además se describieron cada una de las clases del sistema, así como los principios del diseño utilizados.

# CAPÍTULO 4. IMPLEMENTACIÓN

## Capítulo 4. Implementación

### 4.1 Introducción

Teniendo en cuenta los resultados del análisis y el diseño, en este capítulo se dará paso a la implementación de las clases del diseño en términos de componentes. Además se abordan aspectos sobre el tratamiento de errores y la seguridad.

### 4.2 Modelo de implementación

En el flujo de trabajo de implementación se define la organización del sistema en términos de subsistemas de implementación, organizados en capas que describen cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

#### 4.2.1 Diagrama de despliegue

El diagrama de despliegue se utiliza para modelar la vista de despliegue estática de un sistema, muestra las relaciones físicas entre los componentes hardware y software, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software. Este diagrama muestra además los nodos que participan en la ejecución y las relaciones de dependencia y asociación que se establecen entre ellos. Los nodos pueden ser servidores, procesadores o dispositivos.

A continuación se exponen dos diagramas de despliegue, ya que existen dos formas en las que se puede desplegar la aplicación, atendiendo a las necesidades del cliente. El primero está compuesto por dos nodos: uno representa la computadora cliente, donde el usuario hará uso de la aplicación, y el otro representa los servidores web y de BD. El segundo diagrama muestra el servidor web y de BD montado en la misma PC cliente.

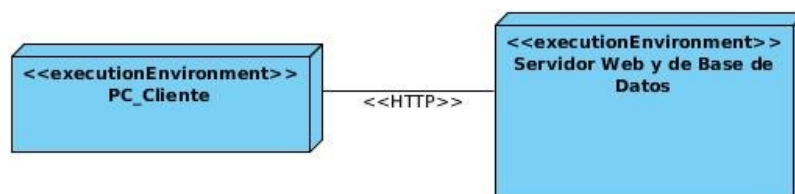


Figura 12: Diagrama de Despliegue 1

## CAPÍTULO 4. IMPLEMENTACIÓN



Figura 13: Diagrama de Despliegue 2

### 4.2.2 Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables.

A continuación se presenta el Diagrama de Componentes General del sistema, en el cual se muestran las relaciones de dependencias entre los paquetes del mismo.

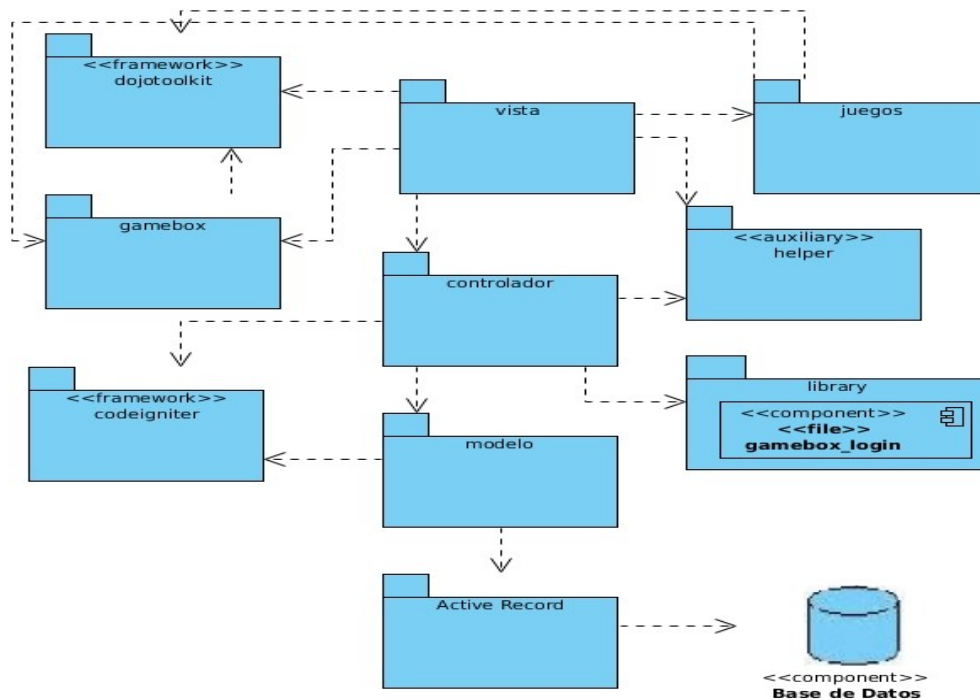


Figura 14: Diagrama de Componentes General

A partir de este diagrama general, se especifica el contenido de los principales paquetes de componentes reflejados en el mismo.

## CAPÍTULO 4. IMPLEMENTACIÓN

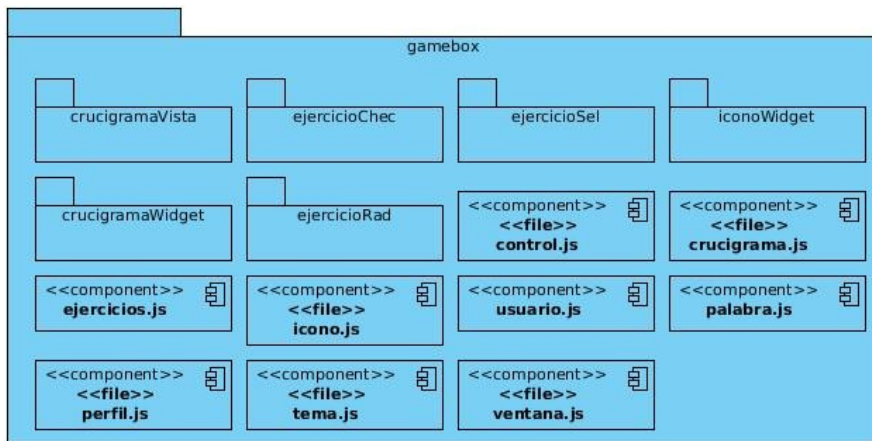


Figura 15: Diagrama de Componentes del paquete gamebox

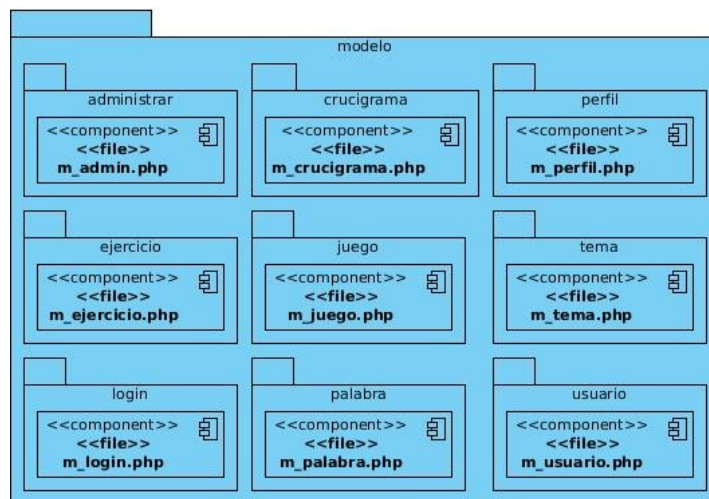


Figura 16: Diagrama de Componentes del paquete modelo

### 4.3 Tratamiento de errores

El tratamiento de errores es una parte esencial de cualquier aplicación, durante la ejecución pueden producirse anomalías y el sistema debe responder adecuadamente a ellas. Entre las ventajas de su aplicación a este producto, se encuentra evitar el envío de formularios con datos incorrectos al servidor. En caso de que suceda esta situación el usuario recibe retroalimentación mediante ventanas de diálogo para que sea capaz de corregir los datos que no cumplan con las especificaciones.

"Una forma efectiva y conveniente de validar (...) es la utilización del *patrón matching*. Algunas personas prefieren llamarlo *expresiones regulares* (...) esta técnica puede ser usada para analizar la entrada del usuario y verificarla siempre que se requiera". [43]



## CAPÍTULO 4. IMPLEMENTACIÓN

### 4.4 Seguridad

Frente a la necesidad de proteger los datos de la aplicación y velar por el correcto uso de la misma se explotan los recursos que provee el framework CodeIgniter. La utilización de Active Record permite que las consultas sean seguras, ya que el sistema escapa cada valor. Por otra parte "la función de filtrado de seguridad se llama automáticamente cuando se invoca un nuevo controlador, ejecutando las siguientes operaciones:

- Provee filtrado XSS (Cross-site Scripting Hacks).
- Destruye el array global GET.
- Destruye todas las variables globales en caso que `register_globals` esté activada.
- Filtra las claves de los arrays POST/COOKIE, permitiendo solamente caracteres alfanuméricos".[44]

El acceso al panel de administración está restringido de forma tal que solo los administradores tengan acceso. Para proteger cada contraseña se utiliza la Clase Encrypt, específicamente el método: `$this->encrypt->sha1()`, función de codificación SHA1 a la cual se le pasa una cadena y "devolverá un hash de una sola vía de 160 bits. SHA1, al igual que MD5, no es decodificable". [44]

### 4.5 Conclusiones

En este capítulo se describe de forma detallada cómo se llevó a cabo el flujo de Implementación descrito por RUP, mediante los diagramas de componentes y despliegue. Del mismo modo se realiza un análisis del tratamiento de errores y los aspectos relacionados con la seguridad, a partir de lo cual se pueden apreciar las características generales del funcionamiento y parámetros tenidos en cuenta en el desarrollo de la aplicación.

## CONCLUSIONES GENERALES

### **Conclusiones generales**

Como resultado del trabajo de investigación realizado, se arribaron a las siguientes conclusiones:

- Se realizó una investigación de aplicaciones similares existentes a nivel internacional, en Cuba y en la UCI, concluyendo que estas presentan algunas características positivas para su utilización; sin embargo no se adecúan a la situación problémica planteada, por lo que se hizo necesario desarrollar una nueva aplicación, donde se estudiaron diferentes tecnologías y herramientas usadas en la actualidad para desarrollar aplicaciones web, seleccionando las más óptimas para su implementación.
- Fueron generados los artefactos fundamentales propuestos por la metodología RUP, obteniendo la documentación necesaria para garantizar el futuro mantenimiento del sistema.
- A partir de la aplicación desarrollada, se obtuvo un producto estable, compatible con otros navegadores y con un backend desde donde pueden gestionarse los contenidos.

## RECOMENDACIONES

### Recomendaciones

Por la importancia que reviste este trabajo de diploma en el quehacer científico-investigativo de la UCI se recomienda:

- Se generalice la propuesta desarrollada en todos los centros educacionales de la enseñanza primaria del país, donde la tecnología disponible lo permita, valiéndose de su flexibilidad para su adaptación a otros contextos educativos.
- Continuar el perfeccionamiento de la propuesta a través de la constante investigación del tema, de forma tal que de respuesta a las necesidades siempre crecientes del sector educacional.
- Incluir nuevos juegos dentro de la aplicación desarrollada.
- Utilizar la información y documentación generada para proyectos futuros y como material de consulta para otras investigaciones relacionadas con el tema.

## REFERENCIAS BIBLIOGRÁFICAS

### Referencias bibliográficas

- 1- **MARQUÉS, Pere.** El Software Educativo. Universidad Autónoma de Barcelona. [Citado el: 1 de Febrero de 2012].
- 2- **SOFTWARE** para la educación ambiental de los estudiantes de secundaria por Ail Sellés Pérez de Villa [et al]. Revista electrónica de la Agencia de Medio Ambiente. 2009, no. 17. [Citado el: 1 de Febrero de 2012]. ISSN 1683-8904
- 3- **AUTO-ENTRENAMIENTO** y auto-evaluación a través de juegos educativos por M. Espinilla [et al]. Jaén, España: Universidad de Jaén, Campus Las Lagunillas. [Citado el: 12 de Enero de 2012.]
- 4- **COLOMA Rodríguez, Orestes.** Hiperentorno de aprendizaje "EUREKA": un software educativo para la enseñanza de la Matemática. Holguín, Cuba: Instituto Superior Pedagógico "José de la Luz y Caballero". [Citado el: 24 de Enero de 2012.]
- 5- **LATORRE, Guillermo.** GCompris: software libre con actividades y juegos educativos para niños. 24 de marzo de 2008. [Citado el: 29 de Enero de 2012.] Disponible en: <http://www.genbeta.com/herramientas/gcompris-software-libre-con-actividades-y-juegos-educativos-para-ninos>
- 6- **SANDRA Quero, Ramones y MADUEÑO Madueño, Leonel.** Sùchiki Walekerù: Un ejemplo del uso de las TIC en escuelas indígenas. Caso Wayuu. Zulia, Venezuela: Universidad de Zulia. 2006. [Citado el: 22 de Febrero de 2012.] ISSN 1316-4910
- 7- Questionmark. Questionmark. [Citado el: 3 de Marzo de 2012.] Disponible en: <http://www.questionmark.com/esp/index.aspx>
- 8- **MARTÍN Pérez, Carlos Juan.** Manual de Usuario. Plataforma de encuestas en línea LimeSurvey. Costa Rica: Universidad Nacional de Costa Rica. Septiembre de 2007. [Citado el: 4 de Febrero de 2012.]
- 9- [Citado el: 10 de Febrero de 2012.] <http://www.aula21.net/segunda/hotpotatoes.htm>
- 10- Crear ejercicios virtuales con EDILIM. Universidad de Salamanca. [Citado el: 1 de Febrero de 2012.] Disponible en: <http://web.usal.es/~angelpoveda/web%20biologia/tutoriales/EDILIM/index.html>
- 11- **PATÓN, D.** Metodologías de Desarrollo de Software. [Citado el: 3 de Marzo de 2012.] Disponible en: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>
- 12- [Citado el: 20 de Febrero de 2012.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>
- 13- RUP vs XP. [Citado el: 20 de Febrero de 2012.] Disponible en: <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>

## REFERENCIAS BIBLIOGRÁFICAS

- 14- **JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James.** El Proceso Unificado de Desarrollo de Software. Madrid: Pearson Education, 2000. [Citado el: 21 de Febrero de 2012.] ISBN 84-7829-036-2.
- 15- **PALACIOS, Juan.** El modelo Scrum. [Citado el: 20 de Febrero de 2012.] Disponible en: [http://eva.uci.cu/file.php/161/Documentos/Materiales\\_complementarios/UD\\_1\\_Procesos/Metodologias/SCRUM/El\\_modelo\\_SCRUM.pdf](http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/SCRUM/El_modelo_SCRUM.pdf)
- 16- **JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James.** 1997. The UML specification documents. Santa Clara, Ck: Rational Software Corp. [Citado el: 4 de Marzo de 2012.] Disponible en <http://www.rational.com>
- 17- **SCHMULLER, Joseph.** Aprendiendo UML en 24 horas. [Citado el: 4 de Marzo de 2012.] Disponible en: <http://orton.catie.ac.cr/cgi-bin/wxis.exe/?IsisScript=AGRIUAN.xis&method=post&formato=2&cantidad=1&expresion=mfn=016171>
- 18- **JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James.** EL Lenguaje Unificado de Modelado. Manual de Referencia. [Citado el: 21 de Febrero de 2012.]
- 19- Herramientas CASE. [Citado el: 18 de Enero de 2012.] Disponible en: <http://www.um.es/docencia/barzana/IAGP/lagp1.html>
- 20- [En línea] [Citado el: 15 de enero de 2012.] <http://www.rational.com.ar/herramientas/roseenterprise.html>
- 21- Revista de Informática Educativa y Medios Audiovisuales [Citado el: 20 de Enero de 2012.] ISSN 1667-8338 Disponible en: <http://laboratorios.fi.uba.ar/lie/Revista/Articulos/080815/A3mar2011.pdf>
- 22- <http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html>
- 23- ¿Qué es HTML? HTML Ya. [Citado el: 20 de Enero de 2012.] Disponible en: <http://www.htmlya.com.ar/temarios/descripcion.php?cod=68&punto=1>
- 24- W3C España. [Citado el: 25 de Enero de 2012.] Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- 25- **EGUÍLUZ Pérez, Javier.** Introducción a JavaScript. 2008. [Citado el: 7 de Febrero de 2012.]
- 26- ¿Qué son los Applets de Java? [Citado el: 10 de Marzo de 2012.] Disponible en: <http://www.desarrolloweb.com/articulos/731.php>

## REFERENCIAS BIBLIOGRÁFICAS

- 27- PHP. [En línea] [Citado el: 15 de Enero de 2012.] Disponible en: <http://www.php.net/manual/es/intro-what-is.php>
- 28- Python. [En línea] 1990. [Citado el: 15 de Enero de 2012.] Disponible en: <http://www.python.org/about/>
- 29- [En línea] [Citado el: 26 de Enero de 2012.] Disponible en: [http://www.r020.com.ar/recursos.php?r\\_id=54&t\\_id=35](http://www.r020.com.ar/recursos.php?r_id=54&t_id=35)
- 30- Eclipse. [En línea] [Citado el: 26 de Enero de 2012.] Disponible en: <http://eclipse.org>
- 31- [En línea] [Citado el: 21 de Febrero de 2012.] Disponible en: <http://www.springframework.org>
- 32- DojoToolkit. [En línea] [Citado el: 5 de Marzo de 2012.] Disponible en: <http://dojotoolkit.org>
- 33- jQuery. [En línea] [Citado el: 5 de Marzo de 2012.] Disponible en: <http://jquery.org>
- 34- CodeIgniter. [En línea] ExpressionEngine. [Citado el: 18 de Enero de 2012.] Disponible en: <http://codeigniter.com/>
- 35- **POTENCIER, Fabien y ZANINOTTO, François.** Symfony 1.2, La guía definitiva. 2008. [Citado el: 18 de Enero de 2012.].
- 36- **GUTIÉRREZ, Juan Diego.** Resumen del libro MySQL 5. [En línea] 2005. [Citado el: 17 de Enero de 2012.]. Disponible en: <http://dialnet.unirioja.es/servlet/libro?codigo=317843>. 978-84-415-1936-7
- 37- **VAZQUEZ Ortiz, Yudisney, CASTILLO Martínez, Gilberto y MIER Pierre, Lisleydi.** Propuesta de un plan de capacitación para la preparación y futura certificación en PostgreSQL. 2011. ISSN: 1994-1536 | RNPS: 0547.
- 38- [En línea] [Citado el: 5 de Marzo de 2012.]. Disponible en: [https://help.ubuntu.com/6.10<apache/HTTPD\\_-\\_Servidor\\_web\\_Apache2/index.html](https://help.ubuntu.com/6.10<apache/HTTPD_-_Servidor_web_Apache2/index.html)
- 39- [En línea] <http://www.ri5.com.ar/ayuda11.php>
- 40- **MARQUINA, Ernesto.** Guía de Patrones, Prácticas y Arquitecturas para .NET 2008 [Citado el: 25 de abril de 2012].
- 41- **PAVÓN, J.** Patrones de diseño orientado a objetos. Patrones estructurales. 2000 [Citado el: 1 de mayo de 2012]. Disponible en: <http://grasia.fdi.ucm.es/jpavon/patrones/patronesestructurales.pdf>
- 42.- **LARMAN, Craig.** UML y Patrones. Introducción al análisis y diseño orientado a objetos. 1999 [Citado el: 12 Abril de 2012] ISBN 970-1 7-0261-1
- 43.- **YUEN K., P. y LAU V.** Practical web technologies. Volumen II. La Habana, Editorial Félix Varela. 2004 [Citado el 7 de mayo 2012].

## REFERENCIAS BIBLIOGRÁFICAS

44.- **VELO, Fernando**. CodeIgniter Guía del Usuario en Español Versión 2.0.2 [En línea]. 2011 [Citado el 7 de mayo 2012]. Disponible en: [http://sunshine.prod.uci.cu/gridfs/sunshine/books/CodeIgniter\\_Guia\\_Usuario.pdf](http://sunshine.prod.uci.cu/gridfs/sunshine/books/CodeIgniter_Guia_Usuario.pdf)