



UNIVERSIDAD DE HOLGUÍN

Facultad de Informática y Matemática

Hiperentorno de aprendizaje para la enseñanza de la historia del taekwondo en la EIDE “Pedro Díaz Coello” de Holguín

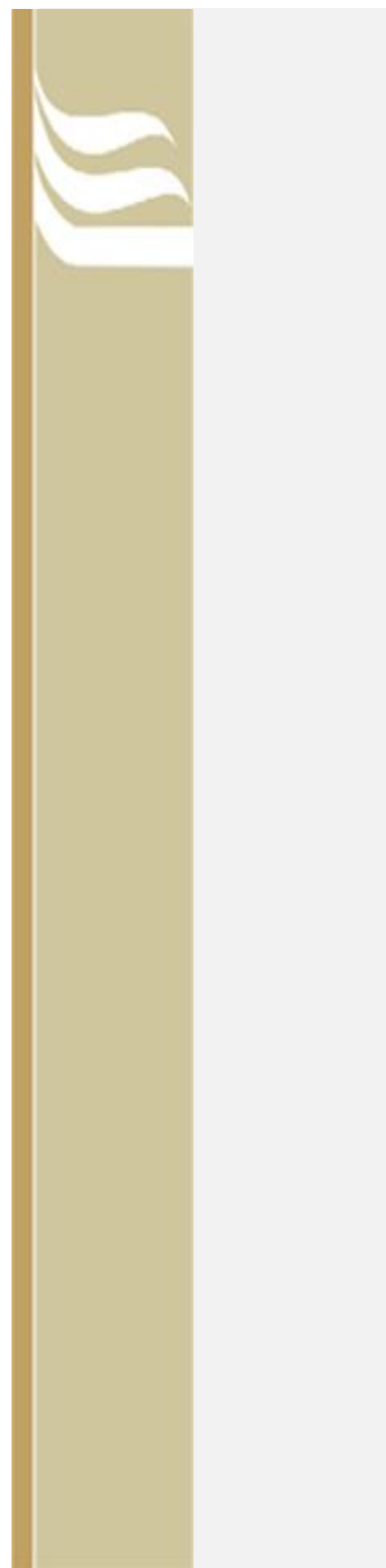
Trabajo de Diploma para optar por el título de Ingeniero Informático

Autor: Luis Enrique Gay Caraballo.

Tutores: M.Sc Jean Carlos Fernández Carralero

M.Sc Sandro Peregrín Carrión

Holguín 2013-2014



Resumen

El enfoque de la gestión del entrenamiento en el Taekwondo, como modalidad de los deportes de combate, intenta la búsqueda de acciones dirigidas a lograr una mayor efectividad y eficiencia de sus direcciones para las diferentes categorías y sexos, teniendo en cuenta que el deporte de alto rendimiento encuentra en el entrenamiento un modo esencial de preparar al deportista para un buen desempeño en la competencia.

Con este fin, se recurre a las Tecnologías de la Información y las Comunicaciones (TIC) y en particular a la informática para desarrollar sistemas de enseñanza asistidos por ordenadores, dirigidos a favorecer el proceso de preparación del deportista y optimizar su auto preparación, apoyados en conceptos, teorías, reglamentos y perfiles de exigencia de la disciplina deportiva, además de contenidos de otras ciencias, necesarios para comprender la esencia del Taekwondo como deporte y arte marcial. En el Catálogo Nacional de Software Educativos del Centro de Información Nacional de Informática del Deporte (CINID) los *software* registrados, en su mayoría, no abordan los contenidos ni las acciones históricas de Taekwondo, refiriéndose solamente en sus contenidos al tratamiento de otras materias docentes como son: Historia, Inglés, Educación Laboral, Informática Básica, Educación Artística, Lengua Española, Literatura y Matemática. En el caso del Taekwondo, existen *software* dirigidos a las evaluaciones de la calidad técnica, sin abordar elementos sobre la preparación histórica en el deporte y reflejándose la falta de preparación histórica por parte de los atletas.

Abstract

The focus of the administration of the training in the Taekwondo, as modality of the combat sports, attempts the search of stocks directed to achieve a bigger effectiveness and efficiency of its addresses for the different categories and sexes, keeping in mind that the high-performance sport finds in the training an essential way of preparing the sportsman for a good acting in the competition.

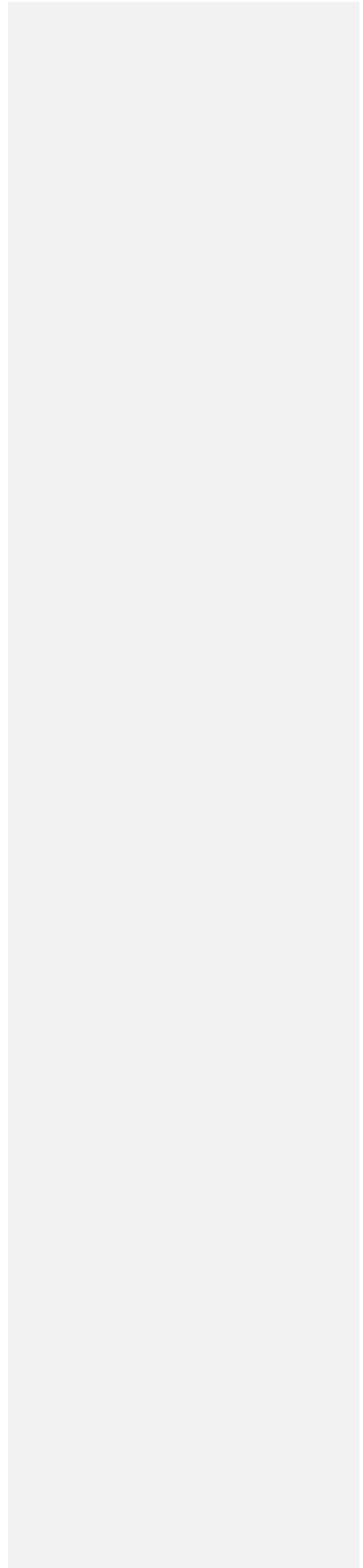
With this end, it is appealed to the Technologies of the Information and the Communications (TIC) and in particular to the computer science to develop education systems attended by computers, directed to favor the process of the sportsman's preparation and to optimize their car preparation, supported in concepts, theories, regulations and profiles of demand of the sport discipline, besides contents of other sciences, necessary to understand the essence of the Taekwondo like sport and martial art. In the Educational National Catalog of Software of the Center of National Information of Computer science of the Sport (CINID) the registered software, in their majority, don't approach the contents neither the historical stocks of Taekwondo, only referring in their contents to the treatment of other educational matters as they are: History, English, Labor Education, Basic Computer science, Artistic Education, Spanish Language, Literature and Mathematics. In the case of the Taekwondo, managed software exist to the evaluations of the technical quality, without approaching elements on the historical preparation in the sport and being reflected the lack of historical preparation on the part of the athletes.

Contenido

Introducción	1
<i>Estructura de la tesis:</i>	7
Capítulo 1.....	8
1.1 Introducción al capítulo.....	8
1.2 Análisis del estado actual del uso de los medios didácticos en la preparación de la dirección teórica del Taekwondo de la EIDE de Holguín	8
1.3 Software Educativo.....	9
1.3.1 Algorítmicos.....	11
1.3.2 Heurísticos.....	12
1.4 Hiperentornos Educativos	13
1.5 Ambientes y plataformas para una gestión eficiente de las redes de conocimientos	14
1.5.1 LCMS (<i>Learning Content Management System</i>) o Sistema Gestor de Contenidos de Aprendizaje.....	15
1.5.2 CMS (<i>Content Management System</i>) o Sistema de Gestión de Contenidos	16
1.6 Otras herramientas de desarrollo.....	18
1.6.1 MySQL	19
1.6.2 Apache Web Server	19
1.7 Metodologías para el desarrollo de <i>software</i>	20
1.7.1 Metodologías tradicionales o pesadas.....	20
1.7.2 Metodologías Ágiles.....	23
1.7.3 UML	33
1.7.4 Visual Paradigm.....	33
Conclusiones del capítulo	34
Capítulo 2.....	35

Introducción	35
2.1 Exploración	35
2.1.1 Descripción de requisitos	35
2.2 Selección de tecnologías y herramientas	40
2.2 Personas relacionadas con la aplicación	41
2.3 Historias de usuarios	41
2.3 Planificación.....	2
2.3.1 Iteraciones	2
2.3.2 Plan de entregas.....	2
2.4 Implementación.....	3
2.5 Valoración de sostenibilidad	10
2.5.1 Dimensión administrativa	10
Dimensión socio-humanista.....	11
2.5.2 Dimensión Ambiental	11
2.5.3 Dimensión tecnológica.....	12
Conclusiones de la valoración de sostenibilidad.....	12
2.6 Valoración de los usuarios	13
2.6.1 Valoración de la metodología mediante criterio de expertos.....	13
Conclusiones del capítulo	15
Conclusiones generales	16
Recomendaciones.....	17
Referencias bibliográficas:.....	¡Error! Marcador no definido.
Bibliografía:.....	¡Error! Marcador no definido.
Anexos	27

Anexo 3. Diagrama de entidad relación.....	35
Anexo 4. Diagrama de clases.....	36
Anexo 5. Diagrama de paquetes.	36



Índice de tablas.

Tabla 1. HU Gestionar historia del Taekwondo	42
Tabla 2. HU Mostrar historia a usuarios	1
Tabla 3. HU Evaluar historia.....	1
Tabla 4. Planificación.....	2
Tabla 5. Plan de entrega	3
Tabla 6. Tarea 1	4
Tabla 7. Tarea 2	4
Tabla 8. Tarea 3	5
Tabla 9. Tarea 4	6
Tabla 10. Tarea 6	6
Tabla 11. Tarea 7	7
Tabla 12. Caso Prueba Aceptación 1.....	8
Tabla 13. Caso Prueba Aceptación 2.....	8
Tabla 14. Caso Prueba Aceptación 3.....	9
Tabla 15. Caso Prueba aceptación 5.....	10

Introducción

La creación del Instituto Nacional de Deporte, Educación Física y Recreación (INDER) el 23 de febrero de 1961, como organismo rector de los procesos dirigidos a fomentar una cultura deportiva en la sociedad cubana, constituyó un momento crucial en el propósito del gobierno revolucionario de convertir al deporte en un derecho del pueblo. Con tal objetivo, se trazaron estrategias para la preparación deportiva del pueblo cubano, a la vez que se recreaba y educaba en las nuevas concepciones y valores de la actividad deportiva.

Cuba se ha convertido en una potencia mundialmente reconocida en los llamados deportes de combate, donde la exigencia y disciplina en el entrenamiento y preparación, tanto por parte de los atletas como de los propios entrenadores y metodólogos, juegan un rol fundamental. Perfeccionar y optimizar los procesos relacionados con dicha preparación, ha sido el propósito elemental de la máxima dirección del deporte en el país. Una de las vías mediante la cual se trabaja en el cumplimiento de este propósito, es precisamente el desarrollo de investigaciones científicas vinculadas directamente a los deportes de combate.

Como necesidad de dichas investigaciones surge, en el año 2002, el Centro de Estudios Nacional de los Deportes de Combate (CENDECOMB), a partir de una iniciativa del Dr.C. Carlos Rafael Núñez Peña[1]. Desde su creación ha tenido como objetivo principal realizar investigaciones en las diferentes esferas de actuación de los deportes de combate, creando proyectos de investigación y desarrollo (I+D), los cuales se sustentan en los cursos de superación permanente, como Diplomados, Maestrías y Estudios de Formación Doctoral, por lo que cada investigador, desde la revisión del banco de problemas de cada deporte, escoge la línea a investigar.

El Taekwondo, cuyo significado literal es el camino del arte de la defensa con las manos y los pies, surge en Corea como una forma de lucha propia, en el período feudal de la dinastía Koguryo. A partir del año 1980 es considerado parte del programa olímpico, cuando fue introducido como deporte de demostración en los

juegos olímpicos de Los Ángeles 84, en los Estados Unidos. Luego de ser introducido en Cuba en el año 1987[2], la calidad de los taekwondistas cubanos comienza a ser mundialmente reconocida por los resultados alcanzados.

La preparación en este deporte está constituida por los siguientes aspectos o direcciones: física, técnica, táctica, psicológica y teórica. En la actualidad, la gestión del entrenamiento está enfocada en la búsqueda de acciones dirigidas a lograr una mayor efectividad y eficiencia de sus direcciones para las diferentes categorías y sexos, teniendo en cuenta que el deporte de alto rendimiento como proceso formador de la personalidad, encuentra en el entrenamiento un modo esencial de preparar al deportista para un buen desempeño en la competencia. En ese sentido, se hacen necesarias las investigaciones que promuevan la idea de reevaluar la concepción del entrenamiento a partir de centrar el interés en la dirección que menos atención tiene por parte de los especialistas, profesionales y profesores deportivos dentro del proceso de preparación del deportista y en especial del taekwondista esta es la dirección teórica y dentro de esta existen diversas subdirecciones (histórica, psicológica, etc.). La presente investigación se centra en la preparación histórica como parte de la dirección teórica del Taekwondo.

Con dicha finalidad, se recurre al empleo de las tecnologías de la información y las comunicaciones (TIC) para desarrollar sistemas de enseñanza asistidos por ordenadores, dirigidos a favorecer el proceso de preparación del deportista y favorecer su auto preparación, apoyados en conceptos, teorías, reglamentos y perfiles de exigencia de la disciplina deportiva, además de contenidos de otras ciencias, necesarios para comprender la esencia del Taekwondo como deporte y arte marcial.

Estos sistemas, cuyos objetivos son estrictamente docentes, son denominados *software* educativos[3], los cuales constituyen programas computacionales cuyas características estructurales y funcionales sirven de apoyo al proceso de enseñar, aprender y administrar[4]. Se puede conceptualizar además como materiales de

aprendizaje especialmente diseñados para ser utilizados con una computadora en los procesos de enseñar y aprender[5].

De acuerdo con las necesidades docentes de los atletas anteriormente planteadas, y el concepto de *software* educativo, la presente investigación se orienta en el empleo de un hiperentorno de aprendizaje. Según el investigador César Labañino Rizzo se le denomina hiperentorno de aprendizaje a *“un sistema informático basado en tecnología hipermedia que contiene una mezcla o elementos representativos de diversas tipologías de software educativo.”*[5] Más adelante afirma que un hiperentorno es *“una mezcla armoniosa de diferentes tipologías de software (tutoriales, entrenadores, simuladores, juegos, etc) sustentada en tecnología hipermedia, concebido para garantizar un apoyo informático a diferentes funciones del proceso de enseñanza aprendizaje, caracterizado fundamentalmente por constituir un apoyo pleno al currículo escolar de un determinado sistema educacional”*[6]. Se define como hiperentorno educativo o hiperentorno de aprendizaje a: un sistema informático basado en tecnología hipermedia, donde se combinan los beneficios de la multimedia y el hipertexto.

El INDER cuenta con el Centro de Información Nacional de Informática del Deporte (CINID), cuyo objetivo es informatizar y apoyar tecnológicamente los procesos referentes a los deportes en Cuba. Dicho centro dispone del Catálogo Nacional de Software Educativos, donde se almacenan productos informáticos registrados que son orientados al apoyo de la labor docente en el deporte.

En los últimos años, la elaboración de *software* educativo se ha favorecido en el país debido a las ventajas que éste representa en el apoyo al proceso de enseñanza – aprendizaje en los distintos renglones de la educación. Sin embargo, en la esfera del deporte, la elaboración y el empleo de estos productos no se han alcanzado los niveles requeridos para satisfacer las necesidades inherentes al proceso de superación, específicamente en los relacionados a la preparación histórica del Taekwondo, a pesar del gran esfuerzo realizado por parte de los

Centros de Estudios del Ministerio de Educación, el CINID y el INDER para desarrollar y emplear estas tecnologías en los centros de educación del deporte.

En estudios realizados se constató que los *software* existentes no son lo suficientemente efectivos para la preparación histórica de los deportistas. Se refieren, en su mayoría, al tratamiento de otras materias, como evaluaciones de la calidad técnica, entre otras, y no abordan contenidos ni hacen referencia a las acciones históricas del Taekwondo.

Uno de los niveles donde la preparación histórica es crítica para la formación integral de atletas, es en las Escuelas de Iniciación Deportiva Escolar (EIDE), donde los deportistas ingresan desde edades tempranas y comienzan su preparación tanto deportiva como docente. La EIDE “Pedro Díaz Coello” de Holguín es una de estas instituciones. Fundada en 1978, tiene como misión encaminar, preparar y forjar los atletas y futuros campeones y glorias deportivas de nuestra provincia, ejerce la enseñanza y la preparación de 25 deportes.

En un estudio preliminar sobre el tema de la preparación histórica del Taekwondo, realizado con la participación de especialistas, técnicos en informática de la EIDE “Pedro Díaz Coello” y del CENDECAMB, se comprobó la existencia de las siguientes limitaciones:

- Poca organización de la bibliografía referente al tema de la historia del Taekwondo, lo que trae como consecuencia la poca preparación en esta materia por parte de los profesores y entrenadores, y por consiguiente, una formación deficiente en los atletas.
- Carencia de medios didácticos para apoyar el proceso de enseñanza – aprendizaje en la dirección histórica del Taekwondo, como materiales audiovisuales y *software* educativos, lo que incide directamente en la preparación de los atletas, específicamente en la apreciación de técnicas de combate y motivación por el estudio de la materia.

- Los *software* educativos existentes no abordan el tema de la historia del Taekwondo con la profundidad requerida.

Tales insuficiencias presuponen la necesidad de ofrecer un tratamiento más profundo al tema, a través del desarrollo de una investigación en la que se le dé respuesta al siguiente **problema científico**: ¿Cómo favorecer el proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín?

En la contextualización de dicho problema se destaca como **objeto de estudio**: el proceso de enseñanza - aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín.

Para dar solución al problema científico se plantea como **objetivo** de la investigación: elaborar un *software* educativo para favorecer el proceso de enseñanza - aprendizaje de la historia del Taekwondo.

Lo anterior delimita el **campo de acción**: la utilización de un *software* educativo en el proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín.

Para dar cumplimiento al objetivo planteado, se consideran oportunas las siguientes **preguntas científicas**:

1. ¿Cuáles son los fundamentos teóricos que sustentan la utilización de un *software* educativo en el proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín?
2. ¿Cuál es el estado actual del proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín?
3. ¿Cómo analizar, diseñar e implementar un *software* educativo para el proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín?

4. ¿Cómo valorar la sostenibilidad de un *software* educativo para el proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín?
5. ¿Cómo valorar la efectividad de la solución propuesta?

Para dar respuestas a las preguntas formuladas se desarrollaron las siguientes **tareas** de investigación:

- Analizar los fundamentos teóricos referentes a la utilización de un *software* educativo en el proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín.
- Diagnosticar el estado actual del proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín.
- Analizar, diseñar e implementar un *software* educativo para el proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín
- Valorar la sostenibilidad de un *software* educativo para el proceso de enseñanza – aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín.
- Valorar la efectividad de la solución propuesta mediante el criterio de expertos.

Métodos teóricos

- Histórico-lógico: para el análisis y determinación de los antecedentes, caracterización y conceptualización del objeto.
- Análisis y síntesis: para la valoración de las diversas fuentes bibliográficas y los resultados obtenidos en la aplicación del juicio de los especialistas.

- Sistémico-estructural: para precisar las características del sistema automatizado a desarrollar, e identificar los elementos que constituyen el objeto de estudio, el campo de acción, así como las relaciones que entre ellos se establecen y lograr la caracterización de los mismos.
- Modelación: en la concepción del sistema automatizado.

Métodos empíricos

- Consulta de documentos: para determinar la información que se utilizará para la confección de este trabajo.
- La entrevista: para la captura de requerimientos y elaboración de los perfiles basados en los usuarios que utilizarán las aplicaciones.
- La encuesta: para la valoración del *software* por parte de un conjunto de expertos.
- Criterio de expertos: para valorar el grado de relevancia de la investigación.

Estructura de la tesis:

La presente investigación está compuesta por introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos. El capítulo uno abarca temas y conceptos referentes al estudio de los *software* educativos, así como al tipo de aplicación a desarrollar, se hace un estudio de las tecnologías y herramientas disponibles para el desarrollo de la solución y se lleva a cabo un diagnóstico referente al objeto en la institución en cuestión. En el capítulo dos se expone la metodología con todas sus fases y en el capítulo tres se procesan y exponen los resultados de las encuestas aplicadas a los expertos.

Capítulo 1

1.1 Introducción al capítulo

En el presente capítulo se exponen los fundamentos teóricos de la investigación, específicamente conceptos relacionados con *software* educativo, se realiza un diagnóstico del estado actual del proceso de enseñanza – aprendizaje de la asignatura y un análisis de las tecnologías y herramientas disponibles.

1.2 Análisis del estado actual del uso de los medios didácticos en la preparación de la dirección teórica del Taekwondo de la EIDE de Holguín

Para llevar a cabo la presente investigación se realizó una serie de entrevistas a 5 profesores que desempeñan funciones metodológicas y administrativas en el CENDECOMB, como el Director de dicho centro, el subdirector, el metodólogo de Taekwondo, entre otros; y que a su vez prestan servicios en la enseñanza del Taekwondo en la EIDE “Pedro Díaz Coello”, específicamente en la dirección teórica.

Al preguntarle qué medios didácticos utilizan para impartir los contenidos referentes a las materias teóricas, dos dijeron que utilizaban la charla; esto confirma la existencia de lagunas en la conceptualización acerca de los medios didácticos, así como su uso, pues la charla no constituye un medio didáctico. Tres profesores deportivos utilizan fotografías y los cinco utilizan videos didácticos de carácter nacional. Al preguntarles sobre su conocimiento acerca de la utilización de *software* educativos que se vincularan con la preparación teórica, los cinco negaron utilizar estos medios en la preparación teórica.

Otro aspecto que se inquirió fueron las necesidades didácticas que presenta la dirección teórica. En dicho aspecto, se mencionaron las siguientes deficiencias:

- Carencia de medios audiovisuales, incluyendo fotos y videos de eventos, atletas y entrenadores internacionales.

- Carencia de bibliografía de la materia
- Deficiente planificación metodológica

Las respuestas a las preguntas formuladas corroboraron la necesidad de reforzar el uso y elaboración de medios didácticos en esta área del entrenamiento del taekwondista.

1.3 Software Educativo

Se define el concepto genérico de software educativo como cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar[4]. Un concepto más restringido de Software Educativo lo define como aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender[7]. Es una aplicación informática, que soportada sobre una bien definida estrategia pedagógica, apoya directamente el proceso de enseñanza aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional del hombre del próximo siglo. Finalmente, los *software* educativos se pueden considerar como el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto del proceso de enseñanza – aprendizaje[3].

Se caracterizan por ser altamente interactivos, a partir del empleo de recursos multimedia, como videos, sonidos, fotografías, diccionarios especializados, explicaciones de experimentados profesores, ejercicios y juegos instructivos que apoyan las funciones de evaluación y diagnóstico [7].

Los *software* educativos pueden tratar las diferentes materias (Matemática, Idiomas, Geografía, Dibujo, etc.), de formas muy diversas, como cuestionarios, simulación de fenómenos, entre otros; y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y más o menos rico en posibilidades de interacción; pero todos comparten las siguientes características:

- Permiten la interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido
- Facilitan las representaciones animadas
- Inciden en el desarrollo de las habilidades a través de la ejercitación
- Permiten simular procesos complejos
- Reducen el tiempo de que se dispone para impartir gran cantidad de conocimientos, facilitando un trabajo diferenciado e introduciendo al estudiante en el trabajo con los medios computarizados
- Facilitan el trabajo independiente y a la vez un tratamiento individual de las diferencias
- Permiten al usuario (estudiante) introducirse en las técnicas más avanzadas

La utilización de los *software* educativos en el proceso de enseñanza-aprendizaje puede ser:

- Por parte del alumno: se evidencia cuando el estudiante opera directamente con el *software* educativo, pero en este caso es de vital importancia la acción dirigida por el profesor.
- Por parte del profesor: se manifiesta cuando el profesor opera directamente con el *software* y el estudiante actúa como receptor del sistema de información. La generalidad plantea que este no es el caso más productivo para el aprendizaje.

El uso del *software* educativo por parte del docente proporciona numerosas ventajas, como son:

- Enriquecer el campo de la pedagogía al incorporar la tecnología de punta que revoluciona los métodos de enseñanza - aprendizaje.

- Constituye una nueva, atractiva y dinámica fuente de conocimientos.
- El docente puede adaptar el *software* a las características y necesidades de su grupo teniendo en cuenta el diagnóstico en el proceso de enseñanza - aprendizaje.
- Permite elevar la calidad del proceso docente - educativo.
- Permite controlar las tareas docentes de forma individual o colectiva.
- Marca las posibilidades para una nueva clase más desarrolladora.

Los *software* educativos a pesar de tener rasgos esenciales básicos y una estructura general común, presentan características diversas en sus interfaces gráficas de usuario de acuerdo al objetivo que cumplen: un laboratorio, una biblioteca, un libro, etc.

Los *software* se pueden clasificar en múltiples tipologías, de acuerdo a diferentes criterios[8].

1.3.1 Algorítmicos

En los *software* educativos de este tipo predomina el aprendizaje vía transmisión del conocimiento, pues el rol del alumno es asimilar el máximo del contenido que se le transmite.

Considerando la función educativa se pueden subclasificar en:

- Sistemas tutoriales: basados en el diálogo con el estudiante, es adecuado para presentar información objetiva, tiene en cuenta las características del alumno, siguiendo una estrategia pedagógica para la transmisión de conocimientos.
- Sistemas entrenadores: se asume que los estudiantes cuentan con los conceptos y destrezas que van a practicar, por lo que su propósito es contribuir al desarrollo de una determinada habilidad, intelectual, manual o

motora, profundizando en las dos fases finales del aprendizaje: aplicación y retroalimentación.

- Libros electrónicos: su objetivo es presentar información al estudiante a partir del uso de texto, gráficos, animaciones, videos, etc., pero con un nivel de interactividad y motivación que le facilite las acciones que realiza.

1.3.2 Heurísticos

Mediante el uso de éstos, el estudiante descubre el conocimiento interactuando con el ambiente de aprendizaje que le permita llegar a él.

Considerando la función educativa se pueden subclasificar en:

- Simuladores: su objetivo es apoyar el proceso de enseñanza – aprendizaje, semejando la realidad de forma entretenida.
- Juegos educativos: ofrecer situaciones excitantes y entretenidas, sin dejar en ocasiones de simular la realidad.
- Sistemas expertos: constituyen un programa de conocimientos intensivos que resuelve problemas que normalmente requieren de la pericia humana. Ejecuta funciones secundarias de manera análoga a un experto, por ejemplo, preguntar aspectos importantes y explicar razonamientos.
- Sistemas tutoriales inteligentes de enseñanza: despiertan mayor interés y motivación, puesto que pueden detectar errores, clasificarlos, y explicar por qué se producen, favoreciendo así el proceso de retroalimentación del estudiante.

En los últimos años ha surgido una nueva tendencia, que es la de integrar en un mismo producto todas o algunas de las tipologías antes expuestas. A este modelo de *software* se le ha denominado Hiperentorno Educativo o Hiperentorno de Aprendizaje.

1.4 Hiperentornos Educativos

El hiperentorno educativo o hiperentorno de aprendizaje, parte de un sistema informático basado en tecnología hipermedia, donde se combinan los beneficios de la multimedia y el hipertexto; por lo cual se puede decir que un hiperentorno de aprendizaje es *“una mezcla armoniosa de diferentes tipologías de software educativo sustentado en tecnología hipermedia, concebido para garantizar un apoyo informático a diferentes funciones del proceso de enseñanza aprendizaje, caracterizado fundamentalmente por constituir un apoyo pleno al currículo escolar de un determinado sistema educacional.”*[3]

El término multimedia se refiere al sistema o proceso que contiene y combina diferentes tipos de medios(media)[9]. La multimedia digital puede comprender animaciones digitales, texto, imágenes fijas, audio digital, video digital e interactividad. El hipertexto constituye un tipo de texto que contiene hipervínculos, los cuales permiten al usuario ir de una página web a otra. El hipertexto organiza la información de forma asociativa en la que una determinada información se fragmenta en una serie de bloques, formalmente denominados nodos[4].

Estos hiperentornos, que tienen en común la utilización de un ambiente informático con una intención didáctica, integran los agentes humanos (estudiantes, profesores) y los artificiales (informáticos) y ofrecen condiciones de interacción local o a través de redes, así como condiciones de acceso a recursos formativos (humanos o mediatizados) locales o distribuidos.

Este tipo de *software* contiene una mezcla de elementos representativos de diversas tipologías de *software* educativo. En los hiperentornos educativos convergen diversas ciencias de forma multidisciplinaria, que organizan colecciones en la esfera del conocimiento, que facilitan un aprendizaje constructivo, de manera colaborativa, con diversidad de estilos en el proceso de formación del educando. En su estructura se aplican diferentes modelos teóricos, de acuerdo al modelo pedagógico utilizado. La idea del hiperentorno educativo se asocia a las redes de conocimiento y la educación a distancia[10].

De un enfoque de trabajo basado en el *software* que estudiaba aspectos específicos y puntuales del proceso de enseñanza – aprendizaje, como la acentuación gráfica en la lengua española, el uso de grafemas y otros, se ha pasado a un enfoque netamente "curricular extensivo", orientado a constituir un soporte informático pleno para los diferentes niveles de enseñanza, sobre la base de series o colecciones que responden a la concepción de "hiperentornos de aprendizaje" en los que se entremezclan diversas tipologías de *software* educativo: tutoriales, entrenadores, simuladores, juegos, etc. en "entornos libres hipermediales"[11].

1.5 Ambientes y plataformas para una gestión eficiente de las redes de conocimientos

Se considera pertinente referirse a la caracterización de las herramientas para la gestión de la formación por medio de redes, la cual comprende:

- Editores de cursos y otros programas para crear materiales multimedia de aprendizaje[13]: *composers* o editores de páginas *web*; programas de autor y programas que permiten la creación de ejercicios de auto evaluación, simulaciones, etc.
- Programas para la comunicación y el trabajo en colectivo, plataformas para el trabajo en colaboración.
- Plataformas de teleformación.
- Plataformas *open source*.
- Programas para la gestión y la administración académica.

En el caso de las plataformas de teleformación pueden clasificarse, a su vez, en: entornos Virtuales de enseñanza-aprendizaje (EVE-A), sistemas telemáticos de teleformación o plataformas de gestión de aprendizaje (*Learning Management System, LMS*).

Sin embargo, la clasificación más conocida y hasta ahora la más utilizada es la de Ferry y Tricia [27], quienes proponen cinco tipos de ambientes en los cuales pueden desarrollarse las labores de educación a distancia:

- *Course Management Systems (CMS)*
- *Synchronous Environments*
- *Learning Management Systems (LMS)*
- *Total Solutions* (una combinación entre *Synchronous Environments*, *CMS* y *LMS*).
- *Related Tools*

Son plataformas en línea que contienen una estructura de soporte para la creación y organización pedagógica de diferentes tipos de materiales didácticos[14]. Es el *software* que, instalado en un servidor *Internet/intranet*, funciona como plataforma para ejecutar, administrar, distribuir y controlar las actividades de formación presenciales o de *e-learning* en la organización. Las plataformas LMS soportan el uso permanente de múltiples usuarios.

1.5.1 LCMS (*Learning Content Management System*) o Sistema Gestor de Contenidos de Aprendizaje

Es un entorno multiusuario en el que los desarrolladores pueden crear, almacenar, reutilizar, gestionar y distribuir contenidos de aprendizaje a partir de un repositorio central de objetos de aprendizaje[4]. En el repositorio, el contenido suele representarse mediante lenguaje XML (*eXtensible Markup Language por sus siglas en ingles*), el cual se ha convertido en un estándar para el intercambio de datos entre aplicaciones de *software*, y especialmente para aplicaciones basadas en *Web*.

La característica esencial para definir a un LCMS es la capacidad de reutilizar el contenido, lo cual se suele apoyar en un modelo de objetos de aprendizaje.

1.5.1.1 Moodle

Moodle es la plataforma open source para eLearning de referencia. Sus capacidades más allá del *eLearning* son muy limitadas. Tiene sentido para implementaciones que requieran cumplir con el protocolo SCORM (*Sharable Content Object Reference Model por sus siglas en inglés*) es una especificación que permite crear objetos pedagógicos estructurados. Con SCORM se hace posible el crear contenidos que puedan importarse dentro de sistemas de gestión de aprendizaje diferentes, siempre que estos soporten la norma SCORM [30].

1.5.2 CMS (Content Management System) o Sistema de Gestión de Contenidos

Es un sistema que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes[15].

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior (directorio) que permite que estos contenidos sean visibles a todo el público (los aprueba).

1.5.2.1 CMS Drupal

Drupal es un sistema de gestión de contenido modular multipropósito muy configurable y a su vez un entorno de desarrollo que permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos[16]. Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web[17].

Es un programa libre, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema [18].

El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet[18]. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar diferentes tipos de sitios web.

Joomla

Joomla es un sistema de gestión de contenidos y un *framework* para aplicaciones web que también puede ser utilizado independientemente. Entre sus principales ventajas está la de permitir editar el contenido de un sitio web de manera sencilla. Es una aplicación de código abierto programada mayoritariamente en PHP bajo una licencia GPL. Este administrador de contenidos puede trabajar en Internet o intranets y requiere de una base de datos MySQL, así como, preferiblemente, de un Apache. En Joomla se incluyen características como: mejora el rendimiento web, versiones imprimibles de páginas, flash con noticias, blogs, foros, *polls* (encuestas), calendarios, búsqueda en el sitio web e internacionalización del lenguaje. Es un producto de software libre. Para el desarrollo de sus múltiples frentes, usa diferentes formas de comunicación como son: los encuentros por

medio de IRC, foros, listas de correo, *wikis* y *blogs*. Este gestor de contenidos sigue las pautas marcadas por el bazar, Su objetivo principal es dar solución a las necesidades de todos aquellos que participan de una u otra forma en el proyecto [28].

WordPress

Es un sistema de gestión de contenido enfocado a la creación de *blogs* (sitios web periódicamente actualizados). Desarrollado en PHP y MySQL, bajo licencia GPL y código modificable. Fue creado a partir del desaparecido *b2/cafelog* y se ha convertido en uno de los principales CMS para la construcción de *blogs*. Mantiene una enorme comunidad de desarrolladores y diseñadores, que se encargan de su desarrollo en general o crear *módulos* y temas para la comunidad [29].

1.6 Otras herramientas de desarrollo

Para la implementación de un hiperentorno de aprendizaje se hace necesaria la utilización de otras herramientas, como los sistemas gestores de bases de datos y los servidores *web*. Por exigencias del cliente se utilizaron el sistema gestor de bases de datos MySQL y como servidor *web* el Apache Web Server.

1.6.1 MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL (del inglés *General Public License*) de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Fue creada por la empresa sueca MySQL AB, que mantiene el *copyright* (derecho de autor) del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es *software* libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un *software* propietario, ya que de no ser así, se vulneraría la licencia GPL.

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API (del inglés *Application Programming Interface*) en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

1.6.2 Apache Web Server

El Apache Server es un servidor HTTP (web) robusto de grado comercial novedoso y de código abierto, disponible y gratis. Es notable por desempeñar un papel fundamental en el crecimiento inicial de la WWW (*World Wide Web*) por sus

siglas en inglés), y ha evolucionado por encima de otros servidores competidores en términos de funcionalidad y rendimiento.

Es desarrollado, mantenido y dirigido por una comunidad abierta de desarrolladores localizados alrededor del mundo.

1.7 Metodologías para el desarrollo de *software*

Las metodologías de desarrollo de *software* establecen un proceso disciplinado sobre el desarrollo del *software* con el fin de hacerlo más predecible y eficiente. Se lleva a cabo a través de un proceso enfocado en planificar, inspirado por otras disciplinas de la ingeniería.

El concepto sobre metodología de desarrollo de *software* se refiere a los métodos, reglas, postulados, procedimientos y procesos que se utilizan para administrar un proyecto de ingeniería de *software*[19]. Estas se dividen en 2 grandes familias: metodologías tradicionales y metodologías ágiles. En los siguientes epígrafes se realiza una comparación entre las dos familias de metodologías, para luego seleccionar la que se utilizará en la confección del sistema en que se desarrolla este trabajo.

1.7.1 Metodologías tradicionales o pesadas

Las metodologías tradicionales o pesadas por su parte son consideradas como la vía habitual para desarrollar *software*. Estas metodologías se basan en una secuencia de pasos, como definición de requerimientos, construcción de la solución, pruebas y despliegue[20]. Estas metodologías requieren definir y documentar un conjunto estable de requerimientos al comenzar el proyecto. Existen varias metodologías pesadas diferentes, donde las más conocidas son: Cascada (*Waterfall* por su traducción en inglés), Modelo en Espiral (*Spiral Model* por su traducción al inglés) y el Proceso Unificado de Desarrollo (*Rational Unified Process* por su traducción al inglés).

Las metodologías pesadas han existido durante un largo tiempo. Ellas imponen un proceso disciplinado en el desarrollo de software con vistas a que este sea más predecible y eficiente. Estos no se han dado a conocer como muy eficientes y menos aún populares, siendo muy criticados por ser metodologías burocráticas, ya que habiendo tanto que seguir en la metodología el proceso de software como un todo se ralentiza.

Existen 4 características comunes de las metodologías pesadas:

1. Enfoque predictivo: las metodologías pesadas tienen la tendencia a planificar primero grandes partes del proceso de software en gran detalle por un largo período de tiempo. Este acercamiento sigue una disciplina de ingeniería donde el desarrollo es predecible y repetitivo.
2. Documentación amplia: el desarrollo de software tradicional mira los requerimientos como una pieza clave de documentación. Un proceso principal en las metodologías pesadas es el gran diseño por adelantado, en el cuál se cree que es posible recoger todos los requerimientos del cliente por adelantado, antes de escribir ningún código.
3. Orientación al proceso: el objetivo de las metodologías pesadas es definir un proceso que trabaje bien para cualquiera que lo utilice. El proceso consistirá en determinadas tareas que deben realizar los administradores, diseñadores, programadores, probadores, etc. Para cada una de estas tareas existe un procedimiento bien definido.
4. Orientación a las herramientas: herramientas de gestión de proyectos, editores de código, compiladores, etc., deben usarse para completar y entregar cada una de las tareas. Las metodologías tradicionales se enfocan más en pensar en el futuro que en entregar software a mediano y corto plazo, como es la necesidad de la mayoría de los usuarios.

Además de estar orientados a grandes proyectos donde se tienen gran cantidad de programadores distribuidos en una serie de roles para lograr la organización del trabajo.

Hoy en día es muy importante tener el trabajo realizado en el plazo más corto posible de tiempo. En este aspecto las metodologías ágiles han ido ganando gran popularidad, incluso en grandes empresas como Microsoft, Nokia y Google.

1.7.1.1 RUP

El Proceso Racional Unificado RUP (*Rational Unified Process* por sus siglas en inglés) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML[21], constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Presenta como principales características:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software
- El RUP es un producto de *Rational* (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

- Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

Algunas desventajas de RUP:

- Si el conjunto de documentos y artefactos no son concebidos tal y como se plantea en RUP, dicha documentación solo servirá para ser archivada lo cual no genera valor respecto a la calidad del desarrollo, y evoluciona en problemas más complejos (aplicación que no satisface los requerimientos, diseño de la estructura no coincide con la estructura final de la aplicación, etc.)
- Es necesario incluir a más personas en el equipo de desarrollo: Especialistas en los diseños y evolución de casos de uso, de los modelos de análisis y diseño, de los modelos de implementación entre otros.

Lo más importante en el desarrollo de un producto informático es el propio desarrollo, en RUP se gasta posiblemente demasiado tiempo para pasar a la fase de desarrollo.

1.7.2 Metodologías Ágiles

El nombre de metodología ágil surge cuando en el año 2001, 17 metodólogos de procesos tuvieron una reunión para discutir tendencias futuras del desarrollo de *software*. En esta reunión firmaron lo que se conoce por el manifiesto ágil, que no es más que los 12 principios básicos de una serie de metodologías que serían conocidas por ágiles para marcar el contraste con las tradicionales o pesadas existentes hasta el momento[20]. Estos principios son los siguientes:

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al período de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

En estos principios se puede observar que los mismos se centran en las personas y no tanto en los procesos, así como en entrega rápida de software y no en complejos diseños previendo el futuro.

1.7.2.1 ICONIX

ICONIX se ubica entre el gran tamaño y complejidad de RUP (Rational Unified Process – Proceso Unificado de Rational) y el pequeño y compacto de XP (eXtreme Programming)[25]. Al igual que RUP, ICONIX es una metodología conducida por casos de uso, pero no incorpora tantos artefactos UML. Es una metodología relativamente pequeña al igual que XP, pero no descarta las etapas de análisis y diseño. Utiliza UML en sus etapas, de modo que se pueden seguir los requerimientos y adaptarse a nuevos cambios.

El objetivo de esta metodología es buscar un subconjunto mínimo, pero suficiente de artefactos UML para hacer un buen trabajo de Ingeniería del Software en poco tiempo.

Las disciplinas que establece ICONIX para el desarrollo de un software:

- Análisis de los Requerimientos
- Análisis y Diseño Preliminar
- Diseño
- Implementación

1.7.2.2 Programación Extrema (XP)

La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad[22]. Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito

del proyecto. Se basa en la realimentación 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[23].

Es una evolución de los problemas causados por los largos ciclos de desarrollo del modelo tradicional. Después de un número exitoso de ensayos en la práctica, la metodología XP fue teorizada en base a las prácticas y principios utilizados.

El ciclo de vida de XP consiste en 6 fases como se pueden observar en la Figura: Exploración, Planeación, Iteraciones con lanzamientos, Producción, Mantenimiento y Muerte.

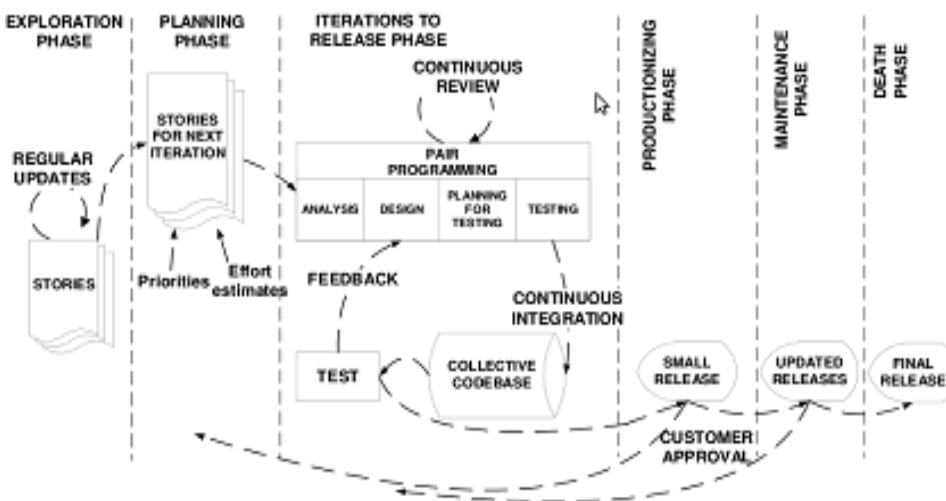


Figura 1. Proceso de desarrollo de XP

A continuación se describe en detalles las funciones de cada una de estas fases:

- **Exploración:** el cliente define las historias de usuario que desea incluir en la primera liberación del producto. Al mismo tiempo los miembros del equipo se familiarizan con las herramientas, tecnologías y prácticas que usarán durante el

proyecto. Se probarán las tecnologías que se usarán y las posibilidades de arquitectura para el sistema, construyendo un prototipo del sistema. La fase de exploración puede tomar desde algunas semanas hasta meses, dependiendo de cuan familiar es la tecnología para los programadores.

- **Planeación:** en esta fase se establece el orden de prioridad para las historias de usuario y un acuerdo entre el cliente y el equipo de desarrollo sobre el contenido del primer lanzamiento del sistema. Los programadores estiman la cantidad de esfuerzo requerido por cada historia de usuario y se llega a un acuerdo en el cronograma del proyecto. Esta fase toma usualmente varios días.

- **Iteraciones a liberar:** esta fase incluye varias iteraciones del sistema antes del primer lanzamiento. El cronograma establecido en la fase de planeación se separa en un número de iteraciones que tomarán de una a cuatro semanas de implementar. La primera iteración crea un sistema con la arquitectura de todo el sistema. Esto se logra seleccionando las historias que fuerzan a construir la estructura de todo el sistema. El cliente decide las historias que serán seleccionadas para cada iteración. Las pruebas funcionales diseñadas por el cliente deben correr al final de cada iteración. Al final de la última iteración el sistema está listo para producción.

- **Producción:** la fase de producción requiere una serie extra de pruebas y el chequeo del funcionamiento del sistema antes de que pueda ser liberado al cliente. Además, en esta fase aún se pueden realizar nuevas decisiones y cambios si se incluyen en el lanzamiento actual. Durante esta fase las iteraciones se deben apresurar desde tres a una semana. Las ideas y sugerencias postpuestas son documentadas para ser implementadas durante la fase de mantenimiento.

- **Mantenimiento:** después del primer lanzamiento de producción para el uso por el cliente, el proyecto XP debe mantener el sistema corriendo y además producir nuevas iteraciones. Para lograr esto, la fase de mantenimiento requiere un esfuerzo en las tareas de soporte del cliente. De esa manera, la velocidad de

desarrollo se puede desacelerar después que el sistema entre en producción. Esta fase puede requerir incorporar nuevas personas al equipo de desarrollo y realizar cambios en la estructura actual del mismo.

- **Muerte:** la fase final o muerte del proceso se encuentra cercana cuando el cliente no tiene más historias para implementar. Esto requiere que el sistema satisfaga otras necesidades del cliente como son rendimiento y confiabilidad. Este es el momento en el que la documentación del sistema es finalmente escrita y no se realizan más cambios al diseño o la arquitectura del proyecto. La muerte del proyecto puede ocurrir también cuando el sistema no entrega el resultado esperado o si se hace muy caro seguir su desarrollo.

Entre sus prácticas encontramos el juego de planeación, pequeños y cortos lanzamientos, metáforas para definir el sistema como un todo, diseño simple, desarrollo dirigido por pruebas, refactorización, programación por parejas, pertenencia colectiva del código, integración continua, semanas de 40 horas, cliente en el sitio de desarrollo, estándares de código, espacio de trabajo abierto y reglas de desarrollo redactadas y aplicadas por el propio equipo.

Las características fundamentales de esta metodología son:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas,** frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación. Véase, por ejemplo, las herramientas de prueba JUnit orientada a Java, DUnit orientada a Delphi y NUnit para la plataforma.NET. Estas dos últimas inspiradas en JUnit.
- **Programación en parejas:** se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y

discutido mientras se escribe es más importante que la posible pérdida de productividad inmediata.

- **Frecuente integración del equipo de programación con el cliente o usuario.** Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores** antes de añadir nueva funcionalidad.
- **Refactorización del código**, es decir, reescribir ciertas partes del código para aumentar su legibilidad y facilidad de mantenimiento pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartida:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad:** Comenzar los proyectos con un diseño simple que evolucione constantemente para añadir la flexibilidad necesaria y elimine la complejidad innecesaria. Una práctica difícil de seguir en XP, debido a que mayormente los programadores tienden a pensar primero en el futuro, lo que entra en contradicción con lo que XP basa sus cimientos: el cambio es inevitable e impredecible.

Todo equipo de desarrollo de software por mucho que se desee auto-organizar debe tener algún tipo de jerarquía para poder funcionar de forma correcta y cada cual tenga una función determinada. En XP existe un grupo de roles que le dan solución a este problema. Es importante señalar que estos roles pueden o no ser exclusivos de una persona y hasta un miembro del equipo puede tener varios roles a la vez. Estos roles son:

- **Programador:** Es el corazón de XP, es la clase trabajadora. Este es el que lleva a cabo el trabajo de codificación y prueba del sistema. Como miembro del equipo de desarrollo puede tomar decisiones de estimación sobre que trabajo aceptar para la iteración y realizar la estimación de este.
- **Ciente:** Este rol cumple con el trabajo como contrapartida del programador. Este es el que decide qué trabajo va a realizar el programador, además de especificar prioridades. La persona que asuma este rol debería trabajar cara a cara con el equipo de desarrollo para que la comunicación entre el cliente y el equipo fluya de forma efectiva y natural.
- **Probador:** Rol de la parte del cliente que no debe confundirse con el que realizan los programadores al realizar desarrollo guiado por pruebas. Este rol se refiere a la parte del cliente que prueba las funcionalidades pactadas entre las partes en el sistema desplegado.
- **Rastreador:** Se considera como el historiador del proyecto, este rol tiene la responsabilidad de llevar el progreso del proyecto, además de asesorar a los programadores en las estimaciones que se realicen. Al analizar el progreso del proyecto en todo su ciclo de vida, se puede predecir comportamientos futuros y la detección de problemas en los equipos de trabajo.
- **Instructor:** Es el responsable de que el proceso de XP se aplique al pie de la letra. El miembro del equipo que asuma este rol debe ser un experto en todo lo relacionado a XP y su aplicación. Además de asesorar a los miembros del equipo para que cumpla con las reglas de este. También debe ser capaz de realizar adaptaciones al proceso que se adecuen mejor a las características del equipo de trabajo.
- **Consultante:** Este rol viene muy útil al proceso de XP cuando se necesite a alguien que tenga un determinado nivel de profundidad en un tema de utilidad para el proyecto. Usualmente al usar programación por parejas

cualquiera de los programadores del equipo puede elegir tomar una determinada tarea. Cuando el conocimiento no es suficiente para completar una tarea, se puede llamar una fuente externa o un propio miembro del equipo con mayor experiencia sobre el tema en cuestión y pedirle que ejerza el rol de consultante.

- **Gerente:** Es el que mantiene al equipo enfocado en una visión general, además de realizar tareas administrativas. Este debe preferentemente tener alguna base de desarrollo de software ya que es el responsable de tomar las decisiones globales del proceso de desarrollo, además de enfocar al equipo en un objetivo general.

La historia es la unidad de funcionalidad en un proyecto XP. El progreso de un proyecto de XP se demuestra a través del código de una historia debidamente probada e integrada. Una historia debe ser entendible tanto por el cliente como por el programador, demostrable a través de algún tipo de prueba, de valor para el primero y lo suficientemente pequeña para que los segundos puedan construir alrededor de media docena de estas en cada iteración.

Las historias deben escribirse por el cliente y definir funcionalidades del sistema en el lenguaje del negocio. En la imagen que se muestra a continuación se pueden ver los campos más comunes en una Historia de Usuario.

Historia de Usuario	
No. Id	Nombre: [Nombre de la Historia]
Usuario: Todos	
Prioridad en el Negocio: [Alta, Media, Baja]	Nivel de Complejidad: [Alta, Media, Baja]
Estimación: [En semanas laborables]	Iteración Asignada: 1
Descripción: [Breve descripción de la historia]	
Información Adicional (Observaciones): [Aquí se pone a qué requisitos de cumplimiento esta historia, entre otras consideraciones]	

Ilustración 1. Historia de usuario

Las historias a su vez se subdividen en tareas más concretas. Estas tareas de carácter técnico, permiten dividir una funcionalidad del negocio en unidades pequeñas estimables por los programadores. Al completar tareas podemos saber el porcentaje del trabajo realizado en cada historia de usuario y llevar trazas de las mismas. Pueden existir además tareas independientes de las historias de usuario que den cumplimiento con una necesidad técnica en específico, como usualmente son las que tienen que ver con el soporte. A continuación se muestra un prototipo de Tarea, las que son escritas por los propios programadores.

Tarea	
Número de Tarea: [#]	Número de la HU: [#]
Nombre de la Tarea: [Nombre de la tarea]	
Tipo de Tarea: [Tipo de Tarea]	Estimación: [Días de duración de la tarea]
Fecha de Inicio: [Fecha de Inicio]	Fecha de Fin: [Fecha de Fin]
Programador Responsable: [Nombre del Programador Responsable]	
Descripción: [Descripción de la Tarea]	

Ilustración 2. Tarea

El otro artefacto usado por XP para documentar el proceso son las Pruebas de Aceptación, las que describen escenarios donde se ponen a prueba funcionalidades del sistema desplegado. Este tipo de pruebas determina cuando las historias de usuario han sido completadas.

1.7.3 UML

UML (*Unified Modeling Language*, Lenguaje Unificado de Modelado) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software[19]. UML es el resultado, en principio, de la unión de los métodos de Booch (*Object Oriented Analysis and Design with Application*) y Rumabugh (*OMT- Object Modeling Technique*) para producir lo que en principio se conoció como el Método Unificado, pero que con la unión de Jacobson (*OOSE-Object Oriented software Engineering: A use case driven approach*) dio paso al Lenguaje Unificado de Modelación. En Noviembre de 1997 este lenguaje (en su versión 1.1) fue adoptado como el estándar por el OMG (*Object Modeling Group*) [19].

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

Para controlar y planificar la propuesta que presenta este trabajo se escogió como metodología el Proceso Unificado de Desarrollo, por sus características y las facilidades.

1.7.4 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML

CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Conclusiones del capítulo

En el presente capítulo se pudo diagnosticar el estado actual del conocimiento de la historia y evolución del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín; se demostró la necesidad de un software educativo para el apoyo del proceso de enseñanza - aprendizaje de la historia del taekwondo; se desarrolló un análisis sobre los diferentes tipos de software educativo, definiciones y características. Se llevó a cabo el análisis de las tecnologías de desarrollo y persistencia a utilizar (Drupal, MySQL, UML, Visual Paradigm), dando razón de la potencialidad que estas ofrecen para el desarrollo de la aplicación escogida.

Capítulo 2

Introducción al capítulo

En el presente capítulo se detallará la aplicación de la metodología XP en la solución propuesta. Se describen las características fundamentales del sistema, exponiéndose los requerimientos a cumplir por el mismo que dan paso a las futuras historias de usuario.

2.1 Exploración

2.1.1 Descripción de requisitos

Los requisitos del sistema representan las tareas que el sistema debe ser capaz de realizar para trabajar correctamente. Es por ello que la redacción detallada de estos, para ser utilizados como base de las futuras pruebas sea de vital importancia.

2.1.1.1 Requerimientos del sistema

En la aplicación se generaliza el término “etapas” ya que es el centro de la aplicación, en acuerdo con el cliente se deciden agrupar todos los términos que a etapas concierne. De tal manera, cada etapa tiene un conjunto de datos que sin estos sería muy pobre la información que se desea ofrecer. Dado estos motivos etapas se define por:

- Atletas
- Árbitros
- Entrenadores
- Eventos
- Reglamentos

- Instalaciones deportivas
- Organizaciones relacionadas
- Limitaciones
- Personalidades

Paquete: administrativo.

1. El sistema debe ser portable.
2. Actualizar períodos de tiempos.
3. Asignar fotos a períodos.
4. Asignar videos a períodos.
5. Asignar documentos a períodos.
6. Actualizar reglamento.
7. Asignar fotos a reglamento.
8. Asignar videos a reglamento.
9. Asignar documentos a reglamento.
10. Actualizar entrenadores.
11. Asignar fotos a entrenadores.
12. Asignar videos a entrenadores.
13. Asignar documentos a entrenadores.
14. Actualizar árbitros.
15. Asignar fotos a árbitros.
16. Asignar videos a árbitros.
17. Asignar documentos a árbitros.
18. Actualizar instalaciones.
19. Asignar fotos a instalaciones.
20. Asignar videos a instalaciones.
21. Asignar documentos a instalaciones.
22. Actualizar eventos.
23. Asignar fotos a eventos.
24. Asignar videos a eventos.

25. Asignar documentos a eventos.

Paquete: Didáctico.

26. Mostrar períodos de tiempos.

27. Mostrar reglamentos.

28. Mostrar entrenadores.

29. Mostrar árbitros.

30. Mostrar instalaciones.

31. Mostrar eventos.

32. Mostrar galería de fotos.

33. Mostrar Galería de videos.

Paquete: evaluativo.

34. Evaluar conocimiento sobre períodos de tiempos.

35. Evaluar conocimiento sobre el reglamento.

36. Evaluar conocimiento sobre los entrenadores.

37. Evaluar conocimiento sobre los árbitros.

38. Evaluar conocimiento sobre las instalaciones.

39. Evaluar conocimiento sobre los eventos.

Paquete de Seguridad.

40. Cerrar sesión.

41. Validar usuario.

42. Autenticar usuario.

43. Actualizar usuario.

44. Visualizar lista de usuarios.

45. Buscar usuario.

46. Cambiar contraseña.

Lista de Reserva del Producto (LRP)

La Lista de Reserva del Producto (LRP) es un documento de alto nivel para el proyecto que contiene amplias descripciones de los requisitos funcionales

ordenados según la prioridad de implementación, ubicada en Muy Alta, Alta, Media y Baja, además de mencionar los requisitos no funcionales del sistema. Cuando un proyecto comienza es muy difícil tener claro todos los requisitos, el equipo de desarrollo se enfoca en las funcionalidades que por el momento suelen ser más importantes para el sistema.

En el transcurso del tiempo esta lista puede crecer y modificarse debido a las constantes reuniones e intercambios entre el cliente y los desarrolladores donde adquieren mayor información sobre el producto. Al terminar de llenar la lista el objetivo es obtener un producto competitivo, lo más correcto posible y que satisfaga las necesidades del cliente.

El uso de esta lista es de gran importancia pues pone en marcha una de las principales características de la metodología XP: ser recomendada para proyectos con requisitos imprecisos y muy variantes. Los programadores que practican esta metodología consideran que los cambios frecuentes de requisitos durante el desarrollo del software son un aspecto natural, inevitable y que poder adaptarse a estos cambios en cualquier parte del ciclo de vida del proyecto es una aproximación mejor y más realista.

No	Funcionalidades del sistema	Prioridad
1	Generar Etapas	Alta
2	Modificar Etapas	Media
2.1	Asignar videos a Etapas	Baja
2.2	Asignar imágenes a Etapas	Baja
2.3	Asignar documentos a Etapas	Baja
3	Mostrar información de las etapas	Alta
4	Evaluar según Etapas	Alta

2.1.1.2 Requerimientos no Funcionales

Los requerimientos no funcionales son aquellos que constituyen propiedades cualidades que el producto debe tener. Son las características que lo hacen atractivo, usable, rápido y confiable.

Apariencia o Interfaz Externa

- Usar interfaces que se adapten a la resolución de la pantalla del usuario.
- Uso de un tema de colores e iconos ligeros y agradables.

Usabilidad

- Lograr que la navegación por la aplicación sea de forma sencilla e intuitiva.

Ayuda y Documentación

- Creación de manual de usuario del producto en un formato estandarizado.

Portabilidad

- La aplicación debe compilarse para las plataformas Win32, Linux x86 y Linux x86_64.

Personalización

- La aplicación debe ser extensible y permitir su personalización.

Uso

- Los usuarios deberán tener preparación en cuanto al uso de las computadoras y su funcionamiento.
- La aplicación contará con un manual de usuario

Seguridad

- Sólo el CENDECOMB estará autorizado a la distribución de la aplicación del sistema diseñado. La información que contiene el sistema estará constantemente bajo la supervisión del CENDECOMB y sólo esta entidad, podrá modificar, eliminar y adicionar nuevos cambios tecnológicos, así como modificaciones de tipo teórica o metodológica. La dirección de las instituciones autorizadas para el uso de la aplicación, será la encargada de determinar el administrador de la aplicación y sus posteriores cambios, previa coordinación con el CENDECOMB.

Software

- Para ejecutar la aplicación, los requerimientos mínimos necesarios son, una computadora con algún servidor web instalado, se recomienda el paquete XAMP, el cual incluye el servidor web Apache y el gestor de bases de datos MySQL, ya configurados e integrados en un ambiente para desarrollar aplicaciones.

Hardware

- Para ejecutar la aplicación los requerimientos mínimos de hardware son:
 - Procesador: Pentium IV para el servidor, Pentium III o superior para los clientes.
 - Red LAN, MAN o WAN
 - Procesador 1200 MHz o superior
 - Memoria 256 MB o superior
 - Disco duro (Mínimo) 80 GB o superior para el servidor, 20 GB ó superior para clientes.
 - Video: 32 MB o superior

2.2 Selección de tecnologías y herramientas

Se tomó como premisa que se deberá seleccionar sólo software libre o de código abierto para apoyar la sostenibilidad del sistema y estar acorde con las políticas trazadas por el país actualmente respecto al tema.

Luego de una revisión de las tecnologías disponibles y la tendencia actual en este tipo de aplicaciones, se decide utilizar como lenguaje de programación PHP. Este lenguaje es el que cumple mejor con los estándares de rendimiento requeridos por la aplicación, además de que su sintaxis se encuentra estandarizada internacionalmente; y existen compiladores libres de gran calidad y en varios sistemas operativos. Como CMS para el desarrollo de la aplicación, se seleccionó Drupal, el cual cuenta con soporte para el lenguaje PHP y HTML, además de diversos módulos y componentes para agilizar los procesos. Para el servidor web se seleccionó Apache, el cual cumple con los requisitos necesarios para el rendimiento esperado del sistema. Como gestor de bases de datos se seleccionó MySQL ya que el mismo cubre las necesidades para lograr una base de datos que no sea pesada y a su vez es firme y perdurable en el tiempo.

Como herramienta CASE se seleccionó el Visual Paradigm, debido a que su calidad es muy superior al resto de los competidores. Para la modelación UML se utilizará la versión desarrollada por la comunidad.

2.3 Personas relacionadas con la aplicación

Las personas relacionadas con el sistema son aquellas que obtienen algún resultado útil del mismo[22]. Las aplicaciones del hiperentorno de aprendizaje de la historia del taekwondo se orientan a especialistas de la rama, como entrenadores y profesores del tema además de otros integradores de la caracterización.

En el caso hiperentorno de aprendizaje de la historia del taekwondo relacionada es el **administrador o especialista y estudiante**. El administrador o especialista debe tener el conocimiento suficiente para administrar el hiperentorno de aprendizaje de la historia del taekwondo.

2.4 Historias de usuarios

Historia de Usuario	
No.: 1	Usuario: Administrador Nombre: Gestionar historia del Taekwondo
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alta
Estimación: 4 semana	Iteración Asignada: 2
Descripción: El <i>especialista</i> debe ser capaz de organizar todo el contenido histórico, por etapa, del Taekwondo.	
Información Adicional (Observaciones): Se le da cumplimiento a los requisitos funcionales 1 y 2 (R1 y R2).	

Tabla 1. HU Gestionar historia del Taekwondo

Historia de Usuario	
No.: 2	Usuario: Todos Nombre: Mostrar historia a usuarios
Prioridad en el Negocio: Media	Nivel de Complejidad: Media
Estimación: 1 semanas	Iteración Asignada: 1
Descripción: El <i>especialista</i> debe ser capaz de mostrar los elementos históricos del taekwondo por etapas.	
Información Adicional (Observaciones): Se le da cumplimiento al requisito funcional 3 (R3).	

Tabla 2. HU Mostrar historia a usuarios

Historia de Usuario	
No.: 3	Usuario: Todos Nombre: Evaluar historia
Prioridad en el Negocio: Alta	Nivel de Complejidad: Alto
Estimación: 4 semanas	Iteración Asignada: 3
Descripción: Realizar evaluaciones al usuario.	
Información Adicional (Observaciones): Con esta historia se da cumplimiento a requisito funcional 4 (R4).	

Tabla 3. HU Evaluar historia

Es importante señalar que las estimaciones del tiempo de duración de cada historia se

utilizaron como unidad la semana laborable que consiste en 5 días, cada uno con 8 horas laborables. Cumpliendo con la práctica de XP que dice que como máximo se deben trabajar 40 horas a la semana[24].

2.3 Planificación

2.3.1 Iteraciones

Para la selección del trabajo de cada iteración se tuvo en cuenta que este no tuviera más de 4 semanas de desarrollo, siguiendo las prácticas básicas de XP. En la Tabla 12 se muestran la distribución de las Historias de Usuario por cada iteración.

Iteraciones	Orden de las historias de usuarios a implementar	Cantidad de tiempo de trabajo
1	Mostrar historia a usuarios	1 semana
2	Gestionar historia del Taekwondo	4 semanas
3	Evaluar historia	4 semanas

Tabla 4. Planificación

2.3.2 Plan de entregas

El plan de entregas es el compromiso final del equipo de trabajo con los clientes[22]. Es una cuestión de vital importancia para el negocio entre ambas partes, ya que la entrega tardía o temprana de la solución, repercute notablemente en la economía y moral de todos los involucrados.

La estimación es uno de los temas más complicados del desarrollo de un proyecto que utiliza la metodología XP, y es por ello que resulta de vital importancia tener bien claros los requerimientos del cliente, el estilo de trabajo del equipo de desarrollo y el tiempo con que dispone el cliente para tener en sus manos la solución.

El cronograma de entregas del Hiperentorno Educativo para la Enseñanza de la Historia del Taekwondo se expone en Tabla 13.

Entregable	Fin 1ra Iteración	Fin 2da Iteración	Fin 3ra Iteración
Hiperentorno educativo para la enseñanza de la historia del taekwondo.	Versión 0.1 Febrero 2013	Versión 0.2 Abril 2013	Versión 1.0 Junio 2013

Tabla 5. Plan de entrega

2.4 Implementación

XP no define de forma obligatoria un conjunto de artefactos a utilizar en la implementación de una solución que utilice dicha metodología[22]. Deja en manos del equipo de desarrollo, dependiendo de sus capacidades de comunicación, la decisión de utilizar tantos tipos de diagramas UML como consideren necesarios, para que el proceso de desarrollo sea más fácil y comprensible.

La metodología de desarrollo XP propone comenzar la implementación del software con una arquitectura sencilla y lo más flexible posible para evitar grandes cambios en próximas iteraciones[25]. A continuación se procede a realizar la primera iteración.

Iteración 1

En esta iteración se persigue como objetivo crear la aplicación inicial con sus funcionalidades básicas, así como la interfaz básica para el trabajo con la misma.

Para realizar la primera iteración se trazaron las siguientes tareas:

1. Tarea #1: Crear estructura del proyecto.
2. Tarea #2: Gestionar etapas.

Tarea	
Número de Tarea: 1	Número de la HU: 1
Nombre de la Tarea: Crear estructura del proyecto	
Tipo de Tarea: Configuración	Estimación: 5 días
Fecha de Inicio: 11/2/2013	Fecha de Fin: 15/2/2013
Programador Responsable: Luis Enrique Gay Caraballo	
Descripción: Crear las funcionalidades necesarias para un funcionamiento básico de la aplicación.	

Tabla 6. Tarea 1

Tarea	
Número de Tarea: 2	Número de la HU: 1
Nombre de la Tarea: Crear funcionalidad básica para seleccionar etapas	
Tipo de Tarea: Desarrollo	Estimación: 5 días
Fecha de Inicio: 18/2/2013	Fecha de Fin: 22/2/2013
Programador Responsable: Luis Enrique Gay Caraballo	

Tabla 7. Tarea 2

Iteración 2

Esta iteración tiene como objetivo crear todas las funciones relacionadas con la selección y modificación de las etapas. Se crearán además todos los componentes

para visualizar la información necesaria.

Para ello se trazaron las siguientes tareas:

2. Tarea #3: Cargar/Modificar elementos de etapas.

Tarea	
Número de Tarea: 4	Número de la HU: 2
Nombre de la Tarea: Cargar/Modificar elementos de etapas.	
Tipo de Tarea: Desarrollo	Estimación: 10 días
Fecha de Inicio: 25/2/2013	Fecha de Fin: 8/3/2013
Programador Responsable: Luis Enrique Gay Caraballo	
Descripción: Implementar las consultas necesarias para la modificación de los datos que lo requieran.	

Tabla 8. Tarea 3

3. Tarea #4 Visualizar elementos de etapas.

Tarea	
Número de Tarea: 5	Número de la HU: 2, 3
Nombre de la Tarea: Visualizar elementos de etapas.	
Tipo de Tarea: Desarrollo	Estimación: 10 días
Fecha de Inicio: 11/3/2013	Fecha de Fin: 21/3/2013
Programador Responsable: Luis Enrique Gay Caraballo	

Descripción: Implementar las consultas necesarias para la visualizar de los elementos que lo requieran.

Tabla 9. Tarea 4

Iteración 3

La tercera iteración persigue uno de los principales objetivos de la aplicación que es la detección automática de **sácadas**. Además para el lanzamiento de la primera versión del producto se quiere elaborar los paquetes de instalación necesarios para cada plataforma.

Comentario [DRIUS1]: ¿?

Para ello se trazaron la siguiente tarea:

2. **Tarea #6:** Efectuar evaluación.

Tarea	
Número de Tarea: 6	Número de la HU: 3
Nombre de la Tarea: Crear modelo de evaluación.	
Tipo de Tarea: Desarrollo	Estimación: 10 días
Fecha de Inicio: 24/3/2013	Fecha de Fin: 5/4/2013
Programador Responsable: Luis Enrique Gay Caraballo	
Descripción: Implementar las funciones necesarias para realizar la evaluación.	

Tabla 10. Tarea 6

Tarea	
Número de Tarea: 7	Número de la HU: 3
Nombre de la Tarea: Efectuar evaluación.	

Tipo de Tarea: Desarrollo	Estimación: 10 días
Fecha de Inicio: 8/4/2013	Fecha de Fin: 19/4/2013
Programador Responsable: Luis Enrique Gay Caraballo	
Descripción: Implementar las funciones necesarias para realizar la evaluación.	

Tabla 11. Tarea 7

2.7 Pruebas

2.7.1 Pruebas de aceptación

Las pruebas de aceptación también son conocidas como pruebas de caja negra, ya que es el propio cliente quien la realiza en compañía de uno de los representantes del equipo de desarrollo y se orientan a las funcionalidades del sistema[22]. Su objetivo es comprobar, desde la perspectiva del usuario final, el cumplimiento de las especificaciones realizadas en las historias de usuario[26].

A continuación, aparecen las pruebas de aceptación realizadas a la solución propuesta:

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: 1
Nombre : Insertar atletas	
Descripción: Prueba la funcionalidad de insertar etapas al sistema.	
Condiciones de ejecución: El usuario debe tener rol de administrador.	
Entrada/Pasos de ejecución: El administrador o profesor encargado una vez dentro de la página de etapas, llena el formulario que se muestra y da clic en	

insertar.
Resultado esperado: La etapa insertada se muestra en la tabla de las etapas registradas, quedando así registrada la etapa deseada.
Evaluación de prueba: Prueba satisfactoria

Tabla 12. Caso Prueba Aceptación 1

Caso de prueba de aceptación	
Código: HU2_P1	Historia de usuario: 2
Nombre : Mostrar etapas	
Descripción: El usuario obtiene las etapas registradas.	
Condiciones de ejecución: Se debe contar con que haya registros de etapas guardados.	
Entrada/Pasos de ejecución: El usuario da clic en el vínculo de “Etapas” y se muestra una tabla con las etapas registradas, su fecha de inicio, su fecha final y una breve descripción.	
Resultado esperado: Muestra una tabla con las etapas registradas, su fecha de inicio, su fecha final y una breve descripción.	
Evaluación de prueba: Prueba satisfactoria.	

Tabla 13. Caso Prueba Aceptación 2

Caso de prueba de aceptación	
Código: HU3_P1	Historia de usuario: 3
Nombre : Realizar evaluación	
Descripción: Prueba la funcionalidad de mostrar una evaluación al usuario acerca del contenido dentro del sistema.	
Condiciones de ejecución: El usuario debe estar registrado.	
Entrada/Pasos de ejecución: El usuario da clic en el hipervínculo de “Evaluación” y se mostrará una lista con las evaluaciones disponibles, en la cual accederá a la evaluación deseada, una vez terminada la evaluación dará clic en enviar y la evaluación se registrara en la base de datos con el nombre del usuario que la envíe.	
Resultado esperado: El formulario de evaluación es enviado con éxito, el sistema muestra un mensaje donde se le informa que se insertó en el registro su evaluación.	
Evaluación de prueba: Prueba satisfactoria	

Tabla 14. Caso Prueba Aceptación 3

Caso de prueba de aceptación	
Código: HU2_P2	Historia de usuario: 2
Nombre : Insertar documentos	
Descripción: Prueba la funcionalidad de subir documentos al sistema.	
Condiciones de ejecución: El usuario debe tener rol de administrador.	
Entrada/Pasos de ejecución: Dentro de cualquier página, da clic en editar y en la casilla de subida de documentos da clic en examinar, busca el documento en la fuente donde se encuentre y da clic en aceptar.	
Resultado esperado: El documento es insertado en el registro del sistema.	
Evaluación de prueba: Prueba satisfactoria.	

Tabla 15. Caso Prueba aceptación 5

2.5 Valoración de sostenibilidad

El desarrollo de todo producto informático implica repercusiones tanto positivas como negativas para los diferentes usuarios que interactuarán con él. Por la razón anterior se hace imprescindible realizar un estudio sobre el impacto que tendrá el producto en las siguientes dimensiones: ambiental, socio humanista, administrativa y tecnológica. Esto garantizará que satisfaga la necesidad que lo origina y las expectativas del cliente.

2.5.1 Dimensión administrativa

Para la valoración de esta dimensión se tiene en cuenta, los gastos implicados para desarrollarla e implantarla, la calidad de la producción y los servicios, la administración de recursos, la toma de decisiones administrativas, la generación de ingresos directos o indirectos a la entidad, etc.

Estas cuestiones están dirigidas a garantizar la sostenibilidad administrativa del sistema. Para el desarrollo de la solución propuesta, no se incurrieron en gastos debido a que el desarrollador es un estudiante.

Comentario [DRIUS2]: Revisar esto con Becerra

Las herramientas utilizadas para el desarrollo de la presente investigación son libres, por lo que no se incurre en gastos para su desarrollo y aplicación. A partir de lo antes analizado y los beneficios que proporciona el modelo, se arribó a la conclusión que esta es sostenible desde la dimensión administrativa.

2.5.2 Dimensión socio-humanista

La dimensión socio-humanista evalúa como el sistema propuesto contribuye con el desarrollo del modo de vida de un grupo social y con la formación ético-humanista de los gestores del proyecto informático, y si satisface una necesidad social.

Con el uso del software, la carga de trabajo se reducirá, al ayudar con las tareas de evaluación de información a los estudiantes. Estas mejoras en las condiciones de trabajo darán lugar a una gran satisfacción del personal implicado, lo que favorecerá una mejor calidad en el proceso cognitivo. A partir de lo analizado anteriormente, se arribó a la conclusión de que el modelo es sostenible desde la dimensión socio-humanista.

2.5.3 Dimensión Ambiental

En la valoración de esta dimensión se tuvo en cuenta, condiciones favorables y no favorables, del o los desarrolladores, minimización de los impactos negativos en la explotación de las tecnologías informáticas. Estos aspectos son tenidos en cuenta para garantizar la sostenibilidad ambiental del modelo propuesto.

En correspondencia con la primera cuestión, el procedimiento se realiza mediante la planificación por etapas, para desarrollar el ambiente de la interfaz se utilizaron colores neutros y suaves, con esto se logra a su vez un estado de ánimo relajado y apto para el aprendizaje por parte del operario del software.

Otras de las cuestiones a tener en cuenta, es que deben de existir como mínimo

algunas condiciones de trabajo para la correcta utilización del procedimiento como son, adecuada iluminación e ubicación del local y las computadoras, asientos cómodos con modelos convenientes para evitar daños en la postura y visualización, ayudando de esta forma a cuidar la salud de la persona. En cuanto al segundo aspecto que se señala, el procedimiento minimiza el impacto negativo de la explotación de las tecnologías, con respecto al medio ambiente. Debido a que el empleo de este modelo reduce el tiempo para impartir los conocimientos y la captación de estos. Tampoco se incurre en gastos de materiales de oficina pues el procedimiento es una solución probada y ejemplificada, lo cual no requiere de gastos innecesarios de papel, tinta y demás, se realizará el proceso desde cero.

En correspondencia con lo anterior expuesto la propuesta tiene un impacto directo favorable en el medio ambiente, por lo cual se valora sostenible en esta dimensión.

2.5.4 Dimensión tecnológica

La valoración, desde este punto de vista, comprende el uso de la tecnología adecuada y asimilable con el diseñador. En este sentido la utilización de tecnologías CMS basadas en software libres, en combinación con un gestor de bases de datos como MySQL hace posible que el costo del procedimiento propuesto sea muy bajo ya que las mismas en su mayoría son herramientas de libre distribución.

En adición a esto la posibilidad de versatilidad de estas herramientas permite que el procedimiento perdure y evolucione en el tiempo. A partir de lo analizado y expuesto anteriormente se arriba a la conclusión de que el procedimiento es sostenible desde la dimensión tecnológica.

2.5.5 Conclusiones de la valoración de sostenibilidad

De acuerdo con los resultados de las dimensiones administrativas en la cual se tuvo en cuenta los beneficios que proporciona la propuesta, se determina sostenible desde las cuatro dimensiones. Logrando que sea perdurable y apto para su explotación libremente por los usuarios interesados.

Debido a que la propuesta responde a una necesidad directa del CENDECOMB a un mínimo costo y su aplicación puede ser extensiva a otros usos no considerados inicialmente podemos concluir que la propuesta es sostenible en las cuatro dimensiones analizadas.

La valoración que da el cliente acerca del producto final también es de vital importancia. A este es al que se destina la aplicación y por lo tanto su opinión es definitiva respecto a la factibilidad de la solución.

La metodología XP ofrece como ya se había visto anteriormente las pruebas de aceptación que diseña y ejecuta el cliente y son aceptadas por el equipo de trabajo como acuerdo de trabajo realizado. Lo que quiere decir que el trabajo se considera completado cuando todas las pruebas de aceptación son satisfactorias para el cliente. En el caso de la solución propuesta, todas las pruebas resultaron satisfactorias para el Cliente.

Por los resultados obtenidos en las dimensiones anteriores se concluye que el software propuesto para la enseñanza de conocimiento es sostenible y perdurable en el tiempo.

2.6 Valoración de los usuarios

Los usuarios quedaron satisfechos con el producto informático final. No se reportaron inconformidades en cuanto a la apariencia general de la aplicación ni a la navegabilidad de la misma y cumple con los requisitos funcionales requeridos por el cliente.

2.6.1 Valoración de la metodología mediante criterio de expertos

Se aplicaron encuestas (Anexo) a un conjunto de personas calificadas como expertos, cuyas áreas de experticia se encuentra relacionada directamente con el tema de la presente investigación. Se seleccionaron 10 expertos, todos graduados de nivel superior, los cuales se clasifican en 7 profesores y 3 entrenadores. La muestra está compuesta por cuatro Doctores en Ciencias, tres Master en Ciencias y tres Licenciados. En la encuesta aplicada se sometieron a evaluación ocho criterios, comprendiendo desde la estructura de la metodología en sí hasta la relevancia de su

aplicación. Cada aspecto fue avalado por los expertos de acuerdo a los siguientes criterios de clasificación: **MR**: muy relevante, **BR**: bastante relevante, **R**: relevante, **PR**: poco relevante y **NR**: no relevante. A continuación, puede aportar una opinión acerca de un elemento que considere destacar.

Conclusiones generales					
Aspectos a Evaluar	MR BR R PR NR				
1. ¿Cómo evalúa la entrada de datos a la aplicación?	Sí	-	-	-	-
2. ¿Cómo evalúa la distribución de los datos en el hiperentorno?	Sí	-	-	-	-
3. ¿Es adecuada la forma de evaluación propuesta?	Sí	-	-	-	-
4. ¿Cómo evalúa la navegación por el sistema?	Sí	-	-	-	-
5. ¿Se cumple con los requisitos de una clase de historia en la actualidad?	Sí	-	-	-	-
6. ¿Cómo evalúa el diseño de las interfaces?	Sí	-	-	-	-
7. ¿Qué tan útil es el producto como herramienta para la enseñanza de la historia del taekwondo?	Sí	-	-	-	-
8. ¿Cómo evalúa la gestión de la información?	Sí	-	-	-	-

Como se observa en la tabla anterior, 5 de los 8 aspectos presentes en la encuesta fueron catalogados como muy relevantes y los otros 3 bastante relevantes, por lo que se asume que todas las fases de la metodología, así como los aspectos que las componen son adecuados.

Conclusiones del capítulo

Se analizó y elaboró el objetivo planteado en la fundamentación de la investigación del proyecto, elaborar un hiperentorno de aprendizaje para la enseñanza de la historia del Taekwondo. El mismo mejorará considerablemente el desempeño y las condiciones de trabajo para el profesor y mejorará además la calidad de las clases, así como se ha de mejorar el proceso de enseñanza.

El sistema no afecta a los usuarios ya que el mismo no genera desempleo ni afecta la salud. Puede ser implantado en los demás deportes con facilidad ya que es de fácil uso y el especialista que lo haga no precisa de profundos conocimientos. Una vez realizada la valoración de sostenibilidad del sistema elaborado, dentro de la cual se consideró la utilización de herramientas y tecnologías de software libre, como una opción viable y superior al software comercial, se puede afirmar que el mismo es sostenible.

Conclusiones generales

Luego de culminar con la presente investigación, se arribaron a las siguientes conclusiones:

- Se realizó un diagnóstico acerca de la situación actual del proceso de enseñanza-aprendizaje de la historia del Taekwondo en la EIDE “Pedro Díaz Coello” de Holguín, evidenciando las deficiencias en dicho proceso.
- Se realizó un estudio de las tecnologías disponibles para el desarrollo del producto informático. Las herramientas y metodologías seleccionadas permitieron una
- Se desarrolló un Hiperentorno de Aprendizaje que contribuyó al estudio de la historia del taekwondo en la EIDE “Pedro Díaz Coello” de Holguín, con características que se adecuan al usuario al que está dirigido.
- Se comprobó, luego de realizar la valoración de sostenibilidad analizando las dimensiones tecnológica, socio-humanista, administrativa y ambiental, que el producto informático desarrollado es sostenible y perdurable en el tiempo.
- Finalizada la implementación de un Hiperentorno de Aprendizaje que contribuya al estudio de la historia del taekwondo en la EIDE “Pedro Díaz Coello” de la ciudad de Holguín la aplicación, se le dio cumplimiento al objetivo planteado al inicio de la presente investigación.

Recomendaciones

Luego de la investigación realizada y teniendo en cuenta los resultados obtenidos, se recomienda:

- Fomentar el uso del mismo por parte de los estudiantes y profesores ya sea mediante clases dirigidas o en el tiempo de máquina y/o estudio individual.
- Ampliar sus funcionalidades con el fin de abarcar, tanto como sea posible, las necesidades educativas que surjan.
- Adaptar el sistema a los otros deportes del centro de estudios.

Referencias bibliográficas:

1. Rodríguez, A., Brea, Lic., TESIS EN OPCION AL TITULO ACADEMICO DE MASTER EN DEPORTES DE COMBATE Teotaek: Software educativo para la preparación de la dirección teórica del entrenamiento a taekwondistas escolares en la Escuela de Iniciación Deportiva Escolar, 2012.
2. Bruceta, O.J., Serrano, Lic . Profesor Auxiliar, SURGIMIENTO Y DESARROLLO DEL TAEKWONDO EN CUBA: EL PROCESO DE INSTITUCIONALIZACION EN HOLGUÍN HASTA 1992, 2009.
3. Coloma, O., Rodríguez, Lic., Prof Aux, M.Sc, Tesis presentada en opción al grado científico de Doctor en Ciencias Pedagógicas CONCEPCIÓN DIDÁCTICA PARA LA UTILIZACIÓN DEL SOFTWARE EDUCATIVO EN EL PROCESO DE ENSEÑANZA APRENDIZAJE, 2008.
4. Ms. Ramos, Lourdes, Pérez, Ms. Domínguez, Junior, Lovaina, Ms. Mariño, Xaily, Gavilondo, and D. Chávez, C. Caridad, Fresno, “¿Software educativo, hipermedia o entorno educativo?,” 2010.
5. Labañino Rizzo, C.y.M.d.T.R., Multimedia para la Educación2001: Ed. Pueblo y Educación, Ciudad de la Habana, 2001.
6. Labañino, R., César y otros., El software educativo en el contexto de la escuela cubana.2007: Ed. Pueblo y Educación.Ciudad de la Habana, 2007.
7. Alvarez, I., Fundora, Lic., Tratamiento de los contenidos de historia de cuba mediante el empleo de los software educativos en la educación primaria, 2013.
8. Ms. Moreno, Carlos, “Software educativos,” 2008. [Online]. Available: <http://educanblog.educando.edu.do/index.php?blogId=1470>.
9. Rodríguez Lamas, R.y.o., Introducción a la Informática Educativa.2002: Ed. Pueblo y Educación, Ciudad de la Habana, 2002.

10. Salas, K., Liudmila, Parra, Trabajo de Diploma para optar por el título de Ingeniero en Informática Hiperentorno de aprendizaje para diseñar actividades de promoción de la lectura, 2008.
11. Curbelo, H., et al., Uso del software educativo en la escuela cubana y su impacto en el aprendizaje de los estudiantes. Curso 40. 2009.
12. Sierra, R., N, Lic., R. Suárez, Cárdenas, Lic., and R. Fndez, Fonseca, MSc, Programa de Preparación del Deportista (Versión IV)2010.
13. Lecourtois, E., Cabrera Uso del software educativo, ¿Necesidad o complacencia?
14. Mulet, M., Angel, Mulet El software educativo: realidades, desafíos y perspectivas.
15. Tem, S., et al. Hiperentornos educativos Educational hypercontexts. 25.
16. T. Tomlinson, Drupal 7, Third Edit. United States of America: Paul Manning, 2010.
17. Gil, F., Experto en Drupal 7. Versión 1. ed2012.
18. Marticorena, J.R., Drupal 7 para usuarios y clientes.
19. S., R., Pressman, Ingeniería de Software. Un enfoque práctico. Quinta edi ed, ed. D. Ince: Mc Graw Hill.
20. Letelier, P. and C. Penadés Metodologías ágiles para el desarrollo de software : eXtreme Programming (XP).
21. Smith, J., A C o m p a r i s o n o f R U P ® a n d X P: Rational Software White Paper,2001.
22. Wallace, B.D., I. Raggett, and J. Aufgang, Extreme Programming for Web Projects2002.

23. Riehle, D., A Comparison of the Value Systems of Adaptive Software Development and Extreme Programming : How Methodologies May Learn from Each Other. 1-13.
24. Joskowicz, I.J. and C. Mingus, Reglas y Prácticas en eXtreme Programming2008. 1-22.
25. Becerra, R., Antonio, García, Trabajo de diploma para optar por el título de *Ingeniero en Informática EYESTRACKER: SOFTWARE PARA EL PROCESAMIENTO DE REGISTROS ELECTRO-OCULOGRÁFICOS DE MOVIMIENTOS OCULARES SACÁDICOS*, 2011.
26. *González, E., Paneque., Trabajo de diploma para optar por el título de Ingeniero en Informática ASISTENTE DE ESCRITORIO PARA LA PLANIFICACIÓN DOCENTE Trabajo*, 2012.
27. *Chris S, Terry M, Tricia. The perfect fit: selecting the online learning environment of tomorrow. 2001. Disponible en: <http://www.deakin.edu.au/itl/research-eval/lms-evaluation/PerfectFit.pdf>*
28. H. Graf, Building Websites with Joomla! Packt Publishing, 2006.
29. R. McCollin and S. Blakeley, Tessa, WordPress Theme Development Beginner 's Guide, Third edit. Packt Publishing Ltd., 2013.
30. N. Smith, Susan and M. Moore, Moodle Course Design Best Practices. Packt Publishing, 2014.

Bibliografía:

- [1] P. G. Torres Lima, Didáctica de las Nuevas Tecnologías de la Información y la Comunicación en Curso 40: Pedagogía 2001. La Habana, 2001.
- [2] R. y otros Rodríguez Lamas, Introducción a la Informática Educativa. Ed. Pueblo y Educación, Ciudad de la Habana, 2002., 2002.
- [3] E. Lecourtois Cabrera, Propuesta Metodológica para el uso del Software Educativo por el profesor de la Secundaria Básica, Tesis en opción al título académico de máster en Investigación Educativa. Guantánamo, 2004.
- [4] C. y otros. Labañino, Rizzo, El software educativo en el contexto de la escuela cubana. Ed. Pueblo y Educación. Ciudad de la Habana, 2007., 2007.
- [5] C. y M. del T. R. Labañino Rizzo, Multimedia para la Educación. Ed. Pueblo y Educación, Ciudad de la Habana, 2001., 2001.
- [6] H. Curbelo, S. E. Rodr, Orestes Coloma, Y. Pe, G. P. Instructora, N. A. Vicedecano, R. Jefe, and S. Educativo, Uso del software educativo en la escuela cubana y su impacto en el aprendizaje de los estudiantes. Curso 40, 2009th ed. Educación Cubana. Ministerio de Educación, 2009, 2009.
- [7] E. La, N.Y. El, H. Desde, A. Ciencias, N. En, E. L. Grado, and O. D. E. La, "SOFTAREAS PARA POTENCIAR EL USO DEL SOFTWARE EDUCATIVO 'LA NATURALEZA Y EL HOMBRE' DESDE LA ASIGNATURA CIENCIAS NATURALES EN EL GRADO OCTAVO DE LA EDUCACIÓN SECUNDARIA BÁSICA."2010.
- [8] Ms. Gutiérrez, José Antonio, Lam, Prof Auxiliar, "El proyecto de Software Educativo. Algunas consideraciones," 2005.

- [9] Ms. Moreno, Carlos, "Software educativos," 2008. [Online]. Available: <http://educanblog.educando.edu.do/index.php?blogId=1470>.
- [10] L. Alvarez, Ileana, Fundora, "Tratamiento de los contenidos de historia de Cuba mediante el empleo de los software educativos en la educación primaria." 2013.
- [11] M. Mulet, Manuel, Angel, "El software educativo: realidades, desafíos y perspectivas." .
- [12] S. Mayet, Moisés Alaín, "Hiperentorno de aprendizaje de didáctica superior." [Online]. Available: <http://www.monografias.com/trabajos-pdf5/hiper-entorno-aprendizaje-didactica-superior/hiper-entorno-aprendizaje-didactica-superior.shtml>.
- [13] M. Mulet, Manuel, Angel, "El Software Educativo. Su influencia en la Enseñanza Primaria." [Online]. Available: <http://www.monografias.com/trabajos82/software-educativo-influencia-ensenanza-primaria/software-educativo-influencia-ensenanza-primaria.shtml> .
- [14] Ms. Ramos, Lourdes, Pérez, Ms. Domínguez, Junior, Lovaina, Ms. Mariño, Xaily, Gabilondo, and D. Chávez, C. Caridad, Fresno, "¿Software educativo, hipermedia o entorno educativo?"
- [15] C. Lecourtois, Ernesto, "Uso del software educativo, ¿Necesidad o complacencia?" .
- [16] L. Castellanos, Kethicer, Rodríguez, "Software educativo." [Online]. Available: <http://www.monografias.com/trabajos31/software-educativo-cuba/software-educativo-cuba.shtml>.
- [17] Ing. Fernández, Jean Carlos, Carralero, "Tesis en opción al título de Master en matemática aplicada e informática para la administración. Metodología de diseño de interfaces gráficas de usuarios para desarrolladores de software de la universidad de Holguín.," Universidad "Oscar Lucero Moya," 2012.

- [18] H. Neyra, Naila, "Software educativo para contribuir al mejoramiento del proceso de enseñanza - aprendizaje de las disciplinas recreativas." .
- [19] L. Mart and L. Guti, "TESIS PRESENTADA EN OPCIÓN AL TÍTULO DE MÁSTER EN CIENCIAS DE LA EDUCACIÓN Procedimiento didáctico para utilizar el hiperentorno prolect en la carrera de ciencias de la información," Universidad "Oscar Lucero Moya," 2012.
- [20] P. Salas, Kenia, Liudmila, "Trabajo de Diploma para optar por el título de Ingeniero en Informática Hiperentorno de aprendizaje para diseñar actividades de promoción de la lectura," Universidad de Holguín "Oscar Lucero Moya," 2008.
- [21] M. S. Coloma, Orestes, Rodríguez, Lic., Prof Aux, "Tesis presentada en opción al grado científico de Doctor en Ciencias Pedagógicas CONCEPCIÓN DIDÁCTICA PARA LA UTILIZACIÓN DEL SOFTWARE EDUCATIVO EN EL PROCESO DE ENSEÑANZA APRENDIZAJE," INSTITUTO SUPERIOR PEDAGÓGICO "JOSÉ DE LA LUZ Y CABALLERO," 2008.
- [22] P. González, Eduardo, "Trabajo de diploma para optar por el título de Ingeniero en Informática ASISTENTE DE ESCRITORIO PARA LA PLANIFICACIÓN DOCENTE Trabajo," Universidad "Oscar Lucero Moya," 2012.
- [23] G. Becerra, Roberto, Antonio, "Trabajo de diploma para optar por el título de Ingeniero en Informática EYESTRACKER: SOFTWARE PARA EL PROCESAMIENTO DE REGISTROS ELECTRO-OCULOGRÁFICOS DE MOVIMIENTOS OCULARES SACÁDICOS," Universidad de Holguín "Oscar Lucero Moya," 2011.
- [24] L. Rodríguez, Armey , Brea, "TESIS EN OPCION AL TITULO ACADEMICO DE MASTER EN DEPORTES DE COMBATE Teotaek: Software educativo para la preparación de la dirección teórica del entrenamiento a taekwondistas escolares

en la Escuela de Iniciación Deportiva Escolar,” UNIVERSIDAD DE CIENCIAS DE LA CULTURA FÍSICA Y EL DEPORTE, 2012.

- [25] D. . Barreto, Iván, Gelles, Titular, Prof., D. . del Toro, Mario, Rodríguez, Titular, Prof., and Ms. Riquenes, Aniuska, Tarragó, TECNOLOGÍA EDUCATIVA : DOS MODELOS PARA LA ACCIÓN DEL MAESTRO, Ministerio. Educación Cubana. Ministerio de Educación, 2009. ISBN, 2009.
- [26] L. Sierra, Roberto, N, L. Suárez, Roberto, Cárdenas, and Ms. Fndez, Roberto, Fonseca, “Programa de Preparación del Deportista (Versión IV),” no. Versión IV. 2010.
- [27] L. . Portilla, Yunior, Rodríguez, Asist., “Tesis en opción al grado científico de Doctor en Ciencias Pedagógicas,” UNIVERSIDAD DE CIENCIAS PEDAGÓGICAS “JOSÉ DE LA LUZ Y CABALLERO” Dirección, 2012.
- [28] Ms. Rodr, Orestes Coloma, A. Men, Y. M. Ricardo, F. Gonz, F. Olaz, M. E. Doallo, A. Z. Ram, Y. P. Rodr, Y. T. Oliva, A. V. Escalona, A. Rodr, R. Ponente, and O. C. Rodr, UN SOFTWARE EDUCATIVO PARA LA ENSEÑANZA DE. MINED, 2006.
- [29] S. Tem, T. Digital, V. Ledo, A. Mar, R. Piedra, P. Auxiliar, E. Nacional, L. Habana, I. I. Grado, L. Habana, and L. Habana, “Hiperentornos educativos Educational hypercontexts,” vol. 25, no. 1, 2010.
- [30] D. Riehle, “A Comparison of the Value Systems of Adaptive Software Development and Extreme Programming : How Methodolo- gies May Learn from Each Other,” no. Xp 2000, pp. 1–13.
- [31] H. Kniberg, Scrum and XP from the Trenches How we do Scrum, © 2007 C4M. C4Media Inc, 2007.

- [32] I. J. Joskowicz and C. Mingus, "Reglas y Prácticas en eXtreme Programming," pp. 1–22, 2008.
- [33] P. Letelier and C. Penadés, "Metodologías ágiles para el desarrollo de software : eXtreme Programming (XP)," Universidad Politécnica de Valencia.
- [34] J. Smith, A Comparison of RUP® and XP. Rational Software White Paper, 2001.
- [35] P. S., Roger, Ingeniería de Software. Un enfoque práctico, Quinta edi. Mc Graw Hill, 2001.
- [36] B. D. Wallace, I. Raggett, and J. Aufgang, Extreme Programming for Web Projects. Addison Wesley, 2002.
- [37] M. C. Solís, "Una explicación de la programación extrema (XP)," 2003.
- [38] L. . P. A. Bruceta, Octavio Jesús , Serrano, "SURGIMIENTO Y DESARROLLO DEL TAEKWONDO EN CUBA: EL PROCESO DE INSTITUCIONALIZACION EN HOLGUÍN HASTA 1992," Instituto Superior de Cultura Física "Manuel Fajardo," 2009.
- [39] F. Gil, Experto en Drupal 7, Versión 1. Forcontu S.L. Todos, 2012.
- [40] O. C. Rodr, D. Mari, B. Lic, Y. P. Rodr, R. Rodr, V. Lic, R. Lic, C. Pedag, C. Holgu, R. Direcci, E. Guti, Y. Lazo, C. Pedag, and P. Tey, EL DESARROLLO DE SOFTWARE EDUCATIVO SIN COSTO DE PROGRAMACIÓN . ¿ UTOPIÍA O REALIDAD ?, 2011th ed. Educación Cubana. Ministerio de Educación, 2011., 2011.
- [41] T. Tomlinson, Drupal 7, Third Edit. United States of America: Paul Manning, 2010.
- [42] R. McCollin and S. Blakeley, Tessa, WordPress Theme Development Beginner ' s Guide, Third edit. Packt Publishing Ltd., 2013.

- [43] B. Aaron, The book you need to succeed! Wiley Publishing, Inc., 2010.
- [44] B. Bondari and E. Griffiths, WordPress 3 Plugin Development Essentials. Packt Publishing Ltd, 2011.
- [45] Jeff Siarto, Head First WordPress. O'REILLY, 2010.
- [46] N. Smith, Susan and M. Moore, Moodle Course Design Best Practices. Packt Publishing, 2014.
- [47] H. Graf, Building Websites with Joomla! Packt Publishing, 2006.

Anexos

Anexo 1. Encuesta a Expertos

Encuesta a Expertos

Nombre y Apellidos: _____

Profesor____ Otras Labores ____

Calificación profesional, grado científico o académico:

Doctor____ Master____ Graduado de nivel superior____

Carrera: _____ Años de experiencia en la profesión: _____

El Centro de Estudio Nacional de Deportes de Combate realiza un estudio para valorar el grado de satisfacción de los usuarios sobre el nuevo hiperentorno educativo de historia del taekwondo. Le agradeceríamos que contestara cuidadosamente el siguiente cuestionario y exprese su criterio. Considere que la opinión que usted aporte contribuirá mucho para nosotros.

Para validar los resultados de la investigación realizada desde el punto de vista técnico emita su criterio valorativo acerca los aspectos que se enuncian en la tabla siguiente utilizando la escala ascendente **MR**: muy relevante, **BR**: bastante relevante, **R**: relevante, **PR**: poco relevante y **NR**: no relevante. A continuación, puede aportar una opinión acerca de un elemento que considere destacar.

Muchas gracias por su cooperación.

Aspectos a evaluar	MR	BR	R	PR	NR
1. ¿Cómo evalúa la entrada de datos a la aplicación?	7	1	0	0	0
2. ¿Cómo evalúa la distribución de los datos en el hiperentorno?	5	2	1	0	0
3. ¿Es adecuada la forma de evaluación propuesta?	6	2	0	0	0
4. ¿Cómo evalúa la navegación por el sistema?	8	0	0	0	0
5. ¿Se cumple con los requisitos de una clase de historia en la actualidad?	3	2	3	0	0
6. ¿Cómo evalúa el diseño de las interfaces?	1	1	6	0	0
7. ¿Qué tan útil es el producto como herramienta para la enseñanza de la historia del taekwondo?	8	0	0	0	0
8. ¿Cómo evalúa la gestión de la información?	7	1	0	0	0

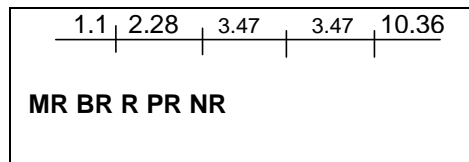
Anexo2. Procesamiento de las encuestas a expertos empleando el método Delphi

TABLA DE FRECUENCIA ABSOLUTA					
Aspectos a evaluar	MR	BR	R	PR	NR
1. ¿Cómo evalúa la entrada de datos a la aplicación?	7	1	0	0	0
2. ¿Cómo evalúa la distribución de los datos en el hiperentorno?	5	2	1	0	0
3. ¿Es adecuada la forma de evaluación propuesta?	6	2	0	0	0
4. ¿Cómo evalúa la navegación por el sistema?	8	0	0	0	0
5. ¿Se cumple con los requisitos de una clase de historia en la actualidad?	3	2	3	0	0
6. ¿Cómo evalúa el diseño de las interfaces?	1	1	6	0	0
7. ¿Qué tan útil es el producto como herramienta para la enseñanza de la historia del taekwondo?	8	0	0	0	0
8. ¿Cómo evalúa la gestión de la información?	7	1	0	0	0

TABLA DE FRECUENCIA ABSOLUTA ACUMULADA					
Aspectos a evaluar	MR	BR	R	PR	NR
1. ¿Cómo evalúa la entrada de datos a la aplicación?	7	8	8	8	8
2. ¿Cómo evalúa la distribución de los datos en el hiperentorno?	5	7	8	8	8
3. ¿Es adecuada la forma de evaluación propuesta?	6	8	8	8	8
4. ¿Cómo evalúa la navegación por el sistema?	8	8	8	8	8
5. ¿Se cumple con los requisitos de una clase de historia en la actualidad?	3	5	8	8	8
6. ¿Cómo evalúa el diseño de las interfaces?	1	2	8	8	8
7. ¿Qué tan útil es el producto como herramienta para la enseñanza de la historia del taekwondo?	8	8	8	8	8
8. ¿Cómo evalúa la gestión de la información?	7	8	8	8	8

TABLA DEL INVERSO DE LA FRECUENCIA ABSOLUTA ACUMULADA					
Aspectos a evaluar	MR	BR	R	PR	NR
1. ¿Cómo evalúa la entrada de datos a la aplicación?	0,875	1	1	1	0,875
2. ¿Cómo evalúa la distribución de los datos en el hiperentorno?	0,625	0,875	1	1	0,625
3. ¿Es adecuada la forma de evaluación propuesta?	0,75	1	1	1	0,75
4. ¿Cómo evalúa la navegación por el sistema?	1	1	1	1	1
5. ¿Se cumple con los requisitos de una clase de historia en la actualidad?	0,375	0,625	1	1	0,375
6. ¿Cómo evalúa el diseño de las interfaces?	0,125	0,25	1	1	0,125
7. ¿Qué tan útil es el producto como herramienta para la enseñanza de la historia del taekwondo?	1	1	1	1	1
8. ¿Cómo evalúa la gestión de la información?	0,875	1	1	1	0,875

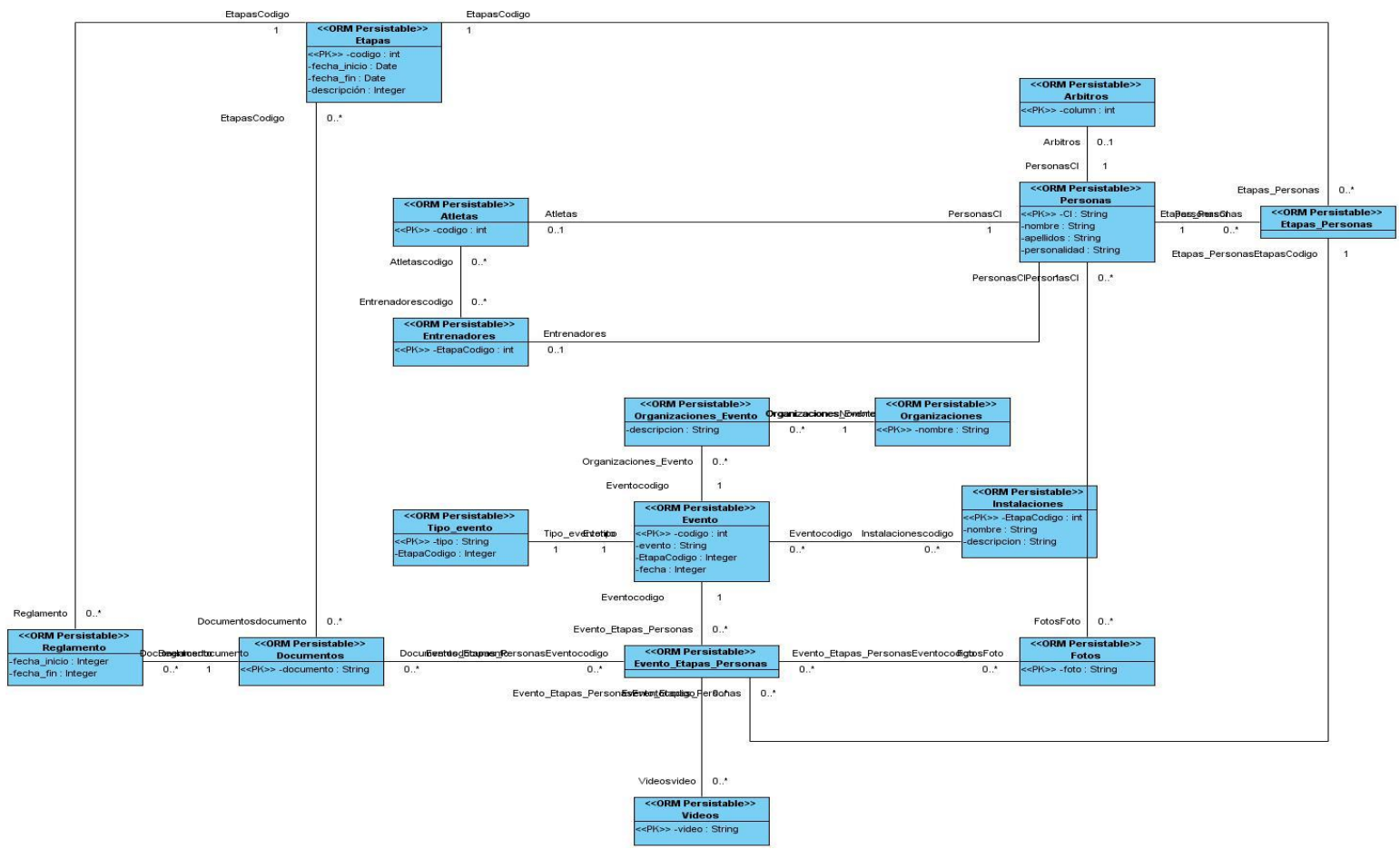
TABLA DE DETERMINACIÓN DE LOS PUNTOS DE CORTES					
Aspectos a evaluar	MR	BR	R	PR	NR
1. ¿Cómo evalúa la entrada de datos a la aplicación?	1,15	3,49	3,49	3,49	11,62
2. ¿Cómo evalúa la distribución de los datos en el hiperentorno?	0,32	1,15	3,49	3,49	8,45
3. ¿Es adecuada la forma de evaluación propuesta?	0,67	3,49	3,49	3,49	11,14
4. ¿Cómo evalúa la navegación por el sistema?	3,49	3,49	3,49	3,49	13,96
5. ¿Se cumple con los requisitos de una clase de historia en la actualidad?	-0,32	0,32	3,49	3,49	6,98
6. ¿Cómo evalúa el diseño de las interfaces?	-1,15	-0,67	3,49	3,49	5,16
7. ¿Qué tan útil es el producto como herramienta para la enseñanza de la historia del taekwondo?	3,49	3,49	3,49	3,49	13,96
8. ¿Cómo evalúa la gestión de la información?	1,15	3,49	3,49	3,49	11,62
Suma	8.8	18.25	27.72	27.72	82.89
Punto de corte (Prom. Columna)	1.1	2.28	3.47	3.47	10.36



Conclusiones generales					
	MR BR R PR NR				
1. ¿Cómo evalúa la entrada de datos a la aplicación?	Sí	-	-	-	-
2. ¿Cómo evalúa la distribución de los datos en el hiperentorno?	Sí	-	-	-	-
3. ¿Es adecuada la forma de evaluación propuesta?	Sí	-	-	-	-
4. ¿Cómo evalúa la navegación por el sistema?	Sí	-	-	-	-
5. ¿Se cumple con los requisitos de una clase de historia en la actualidad?	Sí	-	-	-	-
6. ¿Cómo evalúa el diseño de las interfaces?	Sí	-	-	-	-
7. ¿Qué tan útil es el producto como herramienta para la enseñanza de la historia del taekwondo?	Sí	-	-	-	-
8. ¿Cómo evalúa la gestión de la información?	Sí	-	-	-	-

Conclusiones Generales					
Aspectos a evaluar	MR	BR	R	PR	NR
1. ¿Cómo evalúa la entrada de datos a la aplicación?	Sí	-	-	-	-
2. ¿Cómo evalúa la distribución de los datos en el hiperentorno?	Sí	-	-	-	-
3. ¿Es adecuada la forma de evaluación propuesta?	Sí	-	-	-	-
4. ¿Cómo evalúa la navegación por el sistema?	Sí	-	-	-	-
5. ¿Se cumple con los requisitos de una clase de historia en la actualidad?	Sí	-	-	-	-
6. ¿Cómo evalúa el diseño de las interfaces?	Sí	-	-	-	-
7. ¿Qué tan útil es el producto como herramienta para la enseñanza de la historia del taekwondo?	Sí	-	-	-	-
8. ¿Cómo evalúa la gestión de la información?	Sí	-	-	-	-

Anexo 4. Diagrama de clases.



Anexo 5. Diagrama de paquetes.

