



---

## MÓDULO PARA LA PERSONALIZACIÓN DEL APRENDIZAJE EN LA PLATAFORMA EDUCATIVA MOODLE

---

*Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas*



### **Autores**

Edelin Cruz Febles

Luciano Pumeda Ale

### **Tutores**

MSc. Iván Pérez Mallea

Ing. Leonel Hernández López

La Habana, junio 2016

“Año 58 de la Revolución”

**Declaración de Auditoria**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste, firmamos la presente a los \_\_\_\_ días del mes de julio del año 2016

---

Edelin Cruz Febles

---

Luciano Pumeda Ale

---

Tutor: MSC. Iván Pérez Mallea

---

Tutor: Ing. Leonel Hernández López

## **Resumen**

En la actualidad la educación se ha visto altamente influenciada por el desarrollo de las tecnologías. Términos como educación a distancia, aprendizaje autónomo, móvil o ubicuo son cada vez más comunes, pero ninguno sería posible sin el uso de las tecnologías. Estas han permitido que el aprendizaje se realice de una manera más individual y que los profesores se puedan centrar en las particularidades de sus estudiantes logrando así un mayor grado de personalización del aprendizaje. El objetivo de este trabajo es confeccionar un módulo para la plataforma educativa Moodle que permita a través de técnicas probabilísticas y de análisis de datos mostrar un curso personalizado de acuerdo con la experiencia de trabajo del usuario en la plataforma y los estilos de aprendizaje del mismo.

## Dedicatoria

*De Edelin:*

*A mis abuelos Rosa Castro y Elpidio Febles por todo su amor y dedicación.*

*A mi abuela María Mendoza que aunque no esté entre nosotros siempre estará en mi corazón.*

*A mi mamá linda Lazará Febles por ser mi motor y siempre estar para mí cuando lo necesito.*

*A mi papá Edel Cruz por estar siempre a mi lado de forma incondicional y apoyarme cada vez que lo necesito.*

*A mi hermano Edel Cruz por ser la luz de mis ojos.*

*A mis tíos Tomy Valdés y Roberto Albóniga, por su apoyo, comprensión y dedicación todos estos años.*

*A mi segunda mamá Ida María Irarragory por su amor y por estar siempre preocupada por mí.*

*A mi familia y amigos en general.*

*De Luciano:*

## Agradecimientos

*De Edelin:*

*A mis abuelos: por ser mis segundos padres y aunque todos no estén aquí ahora gracias por estar siempre a mi lado dándome apoyo y amor. Gracias por ayudarme a ser la persona que soy hoy, por no dejarme caer o ayudarme a levantar para hacer mis sueños realidad.*

*A mi mamá por estar siempre a mi lado y confiar en mí cuando más lo necesité, por nunca rendirse conmigo, por ser esa mujer fuerte de la cual vivo orgullosa, por guiarme siempre por el buen camino y sobre todo por ser mi compañera y amiga incondicional.*

*A mi papá que lo adoro por complacerme en todos mis gustos y caprichos según han estado a su alcance, por darme su apoyo y estar conmigo en los momentos buenos y en los no tan buenos también. Te amo, sabes q este título también es tuyo.*

*A mi hermano que siempre me ha apoyado, eres el mejor hermano del mundo. Espero poder ser un ejemplo para ti. A Idita por cuidarme y protegerme siempre; me encanta ser tu niña.*

*A mis tíos por apoyarme en todo momento en especial a mis tíos Tomy y Roberto por acogerme en su casa como a una hija más y brindarme su cariño.*

*A mis primos Yanela, Mayrelis, Miguelito y a todos los que de una forma u otra siempre me han apoyado.*

*A mis amigas Olga, Lizandra por contar con su bella amistad, porque más que amigas son mis hermanas.*

*A mis amigas de apartamento Yaiselin, Tania, Wendy por saber ser mi familia en la UCI cuando los necesite en estos años.*

*A mis tutores Mallea y Leonel por la dedicación brindada para que este trabajo saliera adelante, por ser tan exigentes y preocupados; por las horas y horas que se pasaron conmigo revisando todo. Los dos son muy especiales y nunca los olvidare.*

*A mi compañero de tesis Luciano por aguantarme todo mi estrés en todos los años de carrera, por ser mi paño de lágrimas y mi amigo incondicional.*

*A todos los profes que me han dado su apoyo para que este sueño se hiciera realidad, sin ellos nada de esto hubiera sido posible.*

*A todas aquellas personas que me brindaron su amistad y ayuda siempre que las necesité, que fueron fuente de fuerza y alegría y me impulsaron a seguir adelante hasta alcanzar la meta.*

*De Luciano:*

## Índice

Resumen .....	II
Dedicatoria.....	III
Agradecimientos .....	IV
Introducción .....	1
Capítulo I: Fundamentación teórica.....	7
Introducción .....	7
1.1 – Personalización del aprendizaje.....	7
1.1.1 – Diseño del aprendizaje.....	9
1.1.2 – Modelos de usuario en la personalización del aprendizaje.....	10
1.1.3 – Sistemas de gestión del aprendizaje y Sistemas de Aprendizaje Adaptativo ...	12
1.2 - Plataforma educativa Moodle.....	12
1.2.1 – Arquitectura.....	13
1.2.2 – Extensiones de Moodle (Plugins).....	14
1.2.3 – Rutas de aprendizaje en Moodle.....	16
1.3 – Estudio de sistemas similares .....	16
1.4 – Metodologías .....	21
1.5 – Selección de los lenguajes, herramientas y tecnologías a utilizar .....	25
1.5.1 –Tecnologías para el desarrollo web para el lado cliente .....	25
1.5.2 – Tecnología de desarrollo web para el lado servidor.....	27
1.5.3 – Servidor web .....	27
1.5.4 – Herramienta CASE.....	28
1.5.5 – Entorno de Desarrollo Integrado .....	28

1.6 – Conclusiones .....	29
Capítulo 2: Propuesta de solución.....	30
Introducción .....	30
2.1 – Descripción de la solución.....	30
2.1.1 – Tipos de Actividades y Ponderaciones. ....	32
2.1.2 – Determinación de la experiencia de usuario.....	33
2.1.3 – Determinación del estilo de aprendizaje:.....	36
2.2 – Fase de exploración.....	37
2.3– Personal relacionado con el sistema. ....	37
2.3.1 – Lista de reserva del producto: .....	38
2.3.2 – Aspectos no funcionales del sistema.....	39
2.2.3 – Historias de usuario (HU) .....	40
2.4 – Planificación.....	42
2.4.1– Plan de iteraciones.....	43
2.4.2– Plan de entregas .....	44
2.5– Conclusiones .....	46
Capítulo 3: Diseño, implementación y prueba .....	47
Introducción .....	47
3.1 – Patrón arquitectónico y de diseño .....	47
3.1.2 – Patrones arquitectónicos.....	48
3.1.3 –Patrones de diseño .....	49
3.2 – Diseño del sistema.....	51
3.2.1 – Tarjetas CRC.....	51
3.2.2 – Diagrama de clases del diseño .....	52
3.3 – Diseño de la Base de Datos .....	54



3.3	– Implementación del sistema.....	57
3.4.1	– Tareas de ingeniería.....	57
3.5	– Diagrama de despliegue .....	58
3.6	– Diagrama de componentes .....	60
3.7	– Pruebas .....	61
3.7.1	– Pruebas Unitarias.....	62
3.7.2	– Pruebas de Aceptación .....	63
3.7.3	– Registro de no conformidades.....	64
3.8	– Validación de la investigación.....	66
3.9	– Conclusiones .....	67
	Conclusiones .....	68
	Recomendaciones .....	69
	Referencias.....	70
	Anexos.....	<b>¡Error! Marcador no definido.</b>

## Introducción

La educación desde sus comienzos hasta la actualidad ha experimentado grandes transformaciones y es innegable que estas transformaciones han tenido mayor impacto gracias a la introducción de las tecnologías en el proceso de enseñanza aprendizaje.

Como señalaba David Oakley, en 1983, la transmisión del conocimiento a las siguientes generaciones es una característica de los seres humanos. A lo que Viñes (2008) añade: “Este hecho... nos hace seres en permanente aprendizaje... No podemos evitar aprender. Aprendemos de manera consciente, inconsciente, formal e informal...”.

Esta revolución educacional ha traído tanto a educadores como a alumnos nuevos retos y paradigmas en consecuencia del desarrollo tecnológico y los cambios sociales existentes. Viviendo en una sociedad llena de cambios, con un gran auge de tecnología. Esto trae aparejado que el sistema de enseñanza deba tener mayor énfasis en las características individuales de cada estudiante, pues dicho receptor tiene a su mano diversas formas de aprendizaje y el profesor debe saber brindarle las herramientas que le beneficien más para así obtener una mejor apropiación del conocimiento.

El educar debe estar centrado en un aprendizaje permanente y personalizado para tener éxito y lograr que el educando no pierda el interés de cada materia. Personalizar el aprendizaje es tener en cuenta las individualidades de los estudiantes, este siempre ha sido un precepto de la educación moderna. Con el desarrollo de las tecnologías y su introducción en la educación, con la aparición de la educación semi-presencial, a distancia y ubicua, la atención a estas diferencias individuales se hace más importante debido a que no siempre el estudiante y profesor coinciden en tiempo y lugar.

Un término en el que es interesante detenerse en la actualidad es Entornos Personales de Aprendizaje (PLE, por sus siglas en inglés), el cual consiste en “... el conjunto de herramientas, fuentes y actividades que cada persona utiliza de forma asidua para aprender” (Pérez Mallea, 2015). Se considera que un PLE es el entorno que cada persona crea para favorecer su aprendizaje. Partiendo de esta definición es importante por parte de los estudiantes conocer su manera ideal de instruirse, favoreciendo así la selección de herramientas que se utilizan para aprender. Por parte de los profesores conocer la forma en que sus estudiantes asimilan el

conocimiento facilitándoles así integrar nuevas herramientas y recursos que enriquezcan el proceso de enseñanza aprendizaje; aumentando las diferentes fuentes de consulta, desarrollando la independencia por parte del alumno y favoreciendo la correcta adopción de un PLE.

El centro del ecosistema de aprendizaje de muchas personas son las Plataformas Educativas Virtuales. Estas plataformas permiten al profesor interactuar con sus alumnos mediante foros, promover debates, revisar las tareas orientadas, desarrollar test, entre otras, presentando alternativas en cuanto a las prácticas de educación tradicional. Según Sebastián Días Becerro una Plataforma Educativa Virtual es "... un entorno informático en el que nos encontramos con muchas herramientas agrupadas y optimizadas para fines docentes..." (Becerro, 2009).

Cuba no ha estado exento de los cambios y retos que exige la educación moderna. Se han desarrollado y adoptado diferentes plataformas y soluciones de software como herramientas que contribuyen al desarrollo del conocimiento en los estudiantes.

La UCI (Universidad de las Ciencias Informáticas) desde su inicio marcó el desarrollo del software cubano y sobre todo el del software educativo, desde el centro de desarrollo FORTES (Centro de Tecnologías para la Formación) se han elaborado plataformas educativas, personalizaciones de Sistemas de Gestión del Aprendizaje (Moodle), productos para los diferentes niveles de enseñanza (Multisaber, el Navegante, Colección Futuro, etc.) e incluso videojuegos de corte educativo. Sin embargo, son poco los sistemas que atienden las individualidades de los estudiantes partiendo de diseños de aprendizajes adaptativos y personalizados, aplicados a distintos campos del saber, y en muchos de ellos ligados en extremo a la enseñanza de un dominio del conocimiento y con una estructura monolítica que difícilmente puedan ser extendidos sin elevados esfuerzos de desarrollo (Castillo, 2013).

Durante el año 2013 se desarrolló un módulo para dotar a la colección el Navegante de un sistema de recomendación de ejercicios que fuera capaz de ofrecer nuevos ejercicios en función de la complejidad y los avances que un estudiante muestre durante la utilización del software.

En el año 2015 se realizó un trabajo de diploma titulado "Sistema de Adaptación Inteligente de Contenido para Cursos Virtuales de Aprendizaje" cuyo principal objetivo era "Desarrollar un componente de adaptación inteligente de contenidos que tenga en cuenta los estilos de

aprendizaje de los estudiantes en la plataforma educativa Moodle como forma de tratamiento diferenciado” (Rojas & Alfonso, 2015).

Ninguno de estos productos se centra en adaptar el aprendizaje según las características del estudiante, ni permiten tomar decisiones que modifiquen el diseño instruccional según el desempeño alcanzado a partir de un punto determinado de la trayectoria educativa.

El 25 de junio del 2014 se inauguró el Centro Nacional de Educación a Distancia (CENED) por el Ministro de Educación Superior Dr. Rodolfo Alarcón Ortiz, el cual tiene como sede la UCI. El propósito de este centro es contribuir al desarrollo de programas de enseñanza-aprendizaje en la modalidad virtual, además de ofrecer servicios académicos, consultorías y asesorías en forma no presencial en todo el país y hacia el exterior (Alfonso, 2014).

El CENED en su carácter de rector de la educación a distancia en el país tiene la tarea de llevar todos los adelantos tecnológicos relacionados con la educación a los profesores universitarios de todo el país. Para ello utiliza como plataforma educativa MOODLE (Entorno Aprendizaje Dinámico Modular, Orientando a Objetos), donde se ofrecen cursos a distancia en los que participan profesionales nacionales y extranjeros.

Debido a las características del CENED y las modalidades de enseñanza en las que trabaja, se debe prestar mucha atención a la personalización del aprendizaje ya que todo el trabajo se realiza mediado por la tecnología. Como la interacción es mayoritariamente a través de la plataforma educativa es difícil para el profesor caracterizar a sus estudiantes, sobre todo para determinar estilos de aprendizaje, gustos, desempeño o simplemente para un curso determinado programar diferentes diseños instruccionales encaminados a la atención individual de ciertos estudiantes. Por lo tanto, los cursos se preparan con una variedad de recursos limitada de acuerdo a las experiencias de los profesores impidiendo así que los estudiantes cuenten con variedad de recursos que les permitan aprender de la manera más adecuada a su estilo o preferencias. Los diseños de aprendizaje (diseños instruccionales) son únicos para cada curso obligando a los estudiantes a recibir el contenido de la misma manera. Esto trae como consecuencia que no sientan la plataforma como parte de su ecosistema de aprendizaje o PLE, esto pudiera tener incidencia de manera negativa en el rendimiento académico y contribuir al abandono de los cursos por parte de los mismos.

Ante las necesidades del CENED y teniendo en cuenta todos los elementos mencionados con anterioridad, se plantea el siguiente **problema a resolver**: las limitadas herramientas para la personalización del aprendizaje en los cursos que se ofrecen en la plataforma del CENED no permiten atender las individualidades de los estudiantes.

La investigación se enmarca en el **objeto de estudio**: personalización del aprendizaje en Plataformas Educativas, teniendo como **campo de acción**: personalización del aprendizaje para la plataforma educativa Moodle.

**Objetivo general**: desarrollar un módulo que permita la personalización del aprendizaje en la plataforma educativa del CENED que tenga en cuenta las individualidades del estudiante.

**Objetivos específicos:**

1. Construir el marco teórico referencial de la investigación, donde se relacionan los principales referentes relativos a la personalización del aprendizaje, así como las herramientas y tecnologías para el desarrollo de extensiones en la plataforma educativa Moodle.
2. Desarrollar un módulo para la plataforma educativa Moodle que permita personalizar el aprendizaje de los estudiantes en función de sus individualidades.
3. Realizar pruebas al software como método de validación de las funcionalidades del sistema y verificar el cumplimiento de la hipótesis de trabajo.

**Hipótesis**: el desarrollo de un módulo que tenga en cuenta las individualidades de los estudiantes permitirá personalizar el aprendizaje en la plataforma educativa MOODLE.

**Tareas**

- Determinación de las individualidades de los estudiantes que influyen en la personalización del aprendizaje.
- Realización de un estudio sobre los mecanismos de personalización del aprendizaje.
- Determinación de algoritmos y metodología que sustentan la personalización del aprendizaje.

- Estudio y selección de la metodología de desarrollo de software a utilizar adecuándose a las particularidades del problema a resolver y al propósito del producto final.
- Estudio y selección de los lenguajes de programación a utilizar, así como las herramientas adecuadas para el desarrollo del módulo.
- Aplicación de la metodología seleccionada con el objetivo de construir los artefactos para el desarrollo del módulo de personalización del aprendizaje.
- Implementación dentro de la plataforma Moodle del módulo de personalización del aprendizaje que tenga en cuenta las individualidades y preferencias de los estudiantes.
- Diseño de una estrategia de pruebas para las funcionalidades del módulo de personalización del aprendizaje.
- Realización de pruebas al módulo de personalización del aprendizaje que tenga en cuenta las individualidades y preferencias de los estudiantes.
- Diseño experimental para la verificación de la hipótesis de trabajo y la validación de la investigación.

Con el fin de resolver y dar cumplimiento a los objetivos y las tareas propuestas se emplearon los siguientes **métodos de investigación**:

**Métodos teóricos:**

- Analítico-sintético: se utilizó para el análisis de la teoría, documentación, las partes fundamentales relacionadas con la personalización del aprendizaje y la definición de tecnologías y herramientas más adecuadas para el desarrollo de la propuesta de solución.
- Histórico-Lógico: permitió conocer la evolución de la personalización del aprendizaje en las plataformas educativas, el desarrollo de las extensiones en las diferentes versiones de Moodle, así como las principales características de los sistemas similares.
- Modelación: para el diseño de las pruebas y la validación de la investigación, así como la forma de representación de todos los datos esenciales relacionados con el campo de acción.

**Métodos empíricos:**

- Observación: se utiliza en diferentes momentos de la investigación, posibilitó tomar las mejores prácticas de sistemas similares con el mismo fin de la herramienta a desarrollar.

El documento está organizado en tres capítulos, de la siguiente forma:

*Capítulo I: Fundamentación teórica.*

En este capítulo se expone la fundamentación teórica de esta investigación. Se describen las soluciones informáticas que serán utilizadas (metodologías, modelo de desarrollo de software, tecnologías y herramientas).

*Capítulo II: Propuesta de la solución.*

En este capítulo son descritos los principales conceptos relacionados con la investigación a través de un modelo de dominio. Se identifican los requisitos funcionales y no funcionales. Se elabora la fase de planificación del sistema, donde se crearán las Historias de Usuario y el plan de entrega de cada iteración.

*Capítulo III: Diseño, implementación y prueba.*

En este capítulo se describe la fase de diseño, implementación y prueba. Se identifican y organizan las clases relevantes para las funcionalidades del sistema, las tareas de ingeniería generadas por cada historia de usuario, las tarjetas Clase-Responsabilidad-Colaboración (CRC) así como la arquitectura y los patrones arquitectónicos empleados en el desarrollo de la aplicación. Se describen los principales aspectos del desarrollo, los tipos de pruebas y los casos de prueba realizados al software.

## **Capítulo I: Fundamentación teórica**

### ***Introducción***

En la actualidad existen diversas modalidades de estudio, con una tendencia marcada a la enseñanza flexible y personalizada donde el e-learning<sup>1</sup> adquiere vital importancia. Según el grupo de trabajo de e-Learning 05 de la Red TNet España, lo define como: “conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet / Intranet, que facilitan el acceso a la información y la comunicación con otros participantes” (Aguilar, 2013).

Estas tecnologías de acceso masivo e inclusivo permiten llevar los conocimientos a una gran cantidad de personas. El éxito fundamental radica en los niveles de personalización del aprendizaje que puedan alcanzarse en el diseño de las ofertas formativas utilizando estos medios. En este capítulo se realiza un análisis teórico de las características de la personalización del aprendizaje, las principales metodologías y su implementación en sistemas de gestión del aprendizaje (Learning Management System, LMS, por sus siglas en inglés).

### ***1.1 – Personalización del aprendizaje***

¿Qué es la personalización del aprendizaje a la luz de las tecnologías? Ante los grandes cambios tecnológicos y sociales que atraviesa el mundo es necesario replantear los modelos pedagógicos actuales de modo que contemplen los intereses y particularidades de los estudiantes, con el fin de desarrollar en cada uno de ellos su máximo potencial.

Cada individuo tiene intereses, estilos de aprender y conocimientos previos diferentes. Con esta premisa, el aprendizaje tiene que constituirse en un medio y no un fin en sí mismo. Con el apoyo

---

<sup>1</sup> La traducción literal al español sería aprendizaje electrónico, pero éste no es el término comúnmente utilizado, el término generalizado es e-learning. Existen otros términos en español que se manejan como sinónimos tales como: aprendizaje en red, tele-formación, aprendizaje virtual, educación virtual, formación en línea y enseñanza virtual.



de las Tecnologías de la Información y la Comunicaciones (TIC) es posible lograr un aprendizaje dirigido a explotar estas características particulares de cada individuo, ya que tienen el papel de facilitar el acceso y distribución de los contenidos en red y la interacción entre los participantes. De ese modo, cada estudiante puede gestionar de modo autónomo su propio aprendizaje, según su ritmo e interés. Es ahí donde radica el potencial de las TIC como herramientas que permiten personalizar la enseñanza y transformar a los estudiantes en protagonistas activos de su propio aprendizaje (Prensky, 2016).

La personalización puede establecerse desde varios puntos de vista, pero el más importante es siempre el de los contenidos como punto de partida y los objetivos como meta. Si se elabora contenidos diferenciados, que contemplen y atiendan diferentes enfoques y niveles, se favorece sin duda alguna la adaptación de la formación a las necesidades individuales, permitiendo aprender un mismo contenido respetando las diferentes posibilidades de aprendizaje, los diversos ritmos y los diferentes intereses.

En la sociedad digital actual, el estudiante debe formarse en competencias y habilidades de pensamiento crítico, investigación, análisis, comparación, evaluación, desafío de problemas cotidianos o difíciles desde la toma de decisiones efectivas, estratégicas y oportunas, o mediante la implementación de proyectos. Las TIC ofrecen numerosas herramientas para el desarrollo de estas nuevas habilidades.

Estas competencias le permitirán al estudiante desenvolverse y actuar de modo efectivo, encontrándole el verdadero sentido al aprendizaje. No basta conocer sino saber dirigir el propio aprendizaje para aplicar y construir el conocimiento de modo inteligente y responsable. (Lorenzatti, 2016) .

Claro está que el papel decisivo, y que puede constituir una amenaza, es la complejidad de los diseños de aprendizaje, pues es necesario establecer hitos en el proceso de aprendizaje, diseñar diferentes itinerarios, diversificar los recursos y las actividades, etc. Afortunadamente, los avances de la tecnología del e-learning y de tratamiento de la información hacen que cada vez sea más viable aplicar la personalización del aprendizaje (Sein-Echaluze D. L., 2011).

### **1.1.1 – Diseño del aprendizaje**

Realizar diseños del aprendizaje es la tarea que abordan los docentes de forma continua y, casi rutinaria, cada vez que se enfrentan a un nuevo curso. Un diseño de aprendizaje, como dice Colón (2011), “es un proceso sistemático mediante el cual se analizan las necesidades y metas de la enseñanza y a partir de ese análisis se seleccionan y desarrollan las actividades y recursos para alcanzar esas metas, así como los procedimientos para evaluar el aprendizaje en los alumnos y para revisar toda la instrucción” (Sein-Echaluze & Lerís, 2011).

En la práctica, se dispone de modelos, que guían en la construcción de un diseño de aprendizaje. En ese sentido el modelo más conocido y utilizado en la universidad es el modelo ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación). La etapa, Análisis, consiste en conocer al público al que va dirigida la formación, se centra en conocer los rasgos generales del mismo y en actuar en consecuencia, adaptando la instrucción a esos rasgos.

Según el doctor Miguel Zapata se entiende que el paradigma educativo centrado en el aprendizaje es más ambicioso en el análisis de los estudiantes y señala, como objetivo ideal, el conocer y adaptar el aprendizaje a los rasgos de cada individuo, no a las características que se le presuponen como miembro de un grupo. Es en ese contexto donde la personalización del aprendizaje constituye, de forma incuestionable, una parte sustantiva de los nuevos modelos de aprendizaje e innovación en la práctica instruccional (Zapata-Ros, 2013).

Para la personalización del aprendizaje los profesores deben tener en cuenta varios aspectos (Sein-Echaluze D. L., 2011):

1. Crear opciones a través de un currículo flexible con diversidad de recursos para el mismo contenido.
2. Crear diferentes niveles de contenido o niveles múltiples para que el estudiante pueda aprender a su ritmo y profundizar cuando desee.
3. Crear diferentes momentos de evaluación permitiendo que el estudiante las realice en cualquier momento.
4. Ofrecer la posibilidad de ser utilizado en cualquier lugar y con diversos enfoques para que se acomode a las necesidades de diversos tipos de estudiante.

### **1.1.2 – Modelos de usuario en la personalización del aprendizaje**

El componente fundamental en los sistemas que implementan la personalización del aprendizaje es el llamado modelo de usuario. Con él se pretende recoger y estructurar las variables de personalización del aprendizaje. Las variables consideradas tradicionalmente como características del usuario han sido (Sein-Echaluce D. L., 2011):

- Nivel de conocimientos.
- Estilo de aprendizaje.
- Metas e intereses del usuario.

En el caso del nivel de conocimientos, su determinación es un área de extenso estudio en la que se utiliza el llamado Test Adaptativo Informatizado (TAI). Esta técnica es efectiva como punto de partida para evaluar los conocimientos previos del alumno. Existen otras formas de estimar el nivel de conocimiento según el estudiante avanza en el aprendizaje, pero todas pasan por el sistema de evaluación continua que se diseñe. Muchos sistemas automatizados utilizan los resultados de las evaluaciones como medida del nivel de conocimiento de un alumno.

Como existen diversas formas de educar también existen diversas formas de aprender o como también suelen ser llamados “Estilos de aprendizaje”. Según el psicólogo David Kolb los estilos de aprendizaje son "algunas capacidades de aprender que se destacan por encima de otras como resultado del aparato hereditario de las experiencias vitales propias y de las exigencias del medio ambiente actual." (Universidad de Deusto, 2000 )

La psicóloga Celia Rodríguez Ruiz escribió acerca de los estilos de aprendizaje: “no son más que las diferentes maneras de percibir, organizar y asimilar la información y los conceptos durante las vivencias en las que se construyen aprendizajes” (Celia Rodríguez Ruiz, 2014). Toda persona no aprende de igual manera, así que cada una desarrolla su propio estilo de aprendizaje. En este caso, Celia Rodríguez Ruiz plantea cuatro estilos de aprendizaje los cuales se muestran en la siguiente figura:



Fig. 1 – Estilos de aprendizaje

A pesar que no haya un estilo más eficiente que el otro es bueno conocer cuál es el que utilizan los estudiantes para así poder crear el método de aprendizaje más conveniente con su estilo.

En el trabajo “Adaptive Hypermedia, User Modeling and User-Adapted Interaction” Brusilovsky plantea que un sistema que adapte el aprendizaje debe tener en cuenta no solo los datos del usuario sino también los datos de uso y del entorno. El modelo de usuario propuesto por Brusilovsky cumple con estos aspectos y clasifica las variables de personalización en tres categorías (Brusilovsky, 2001):

- Los datos del usuario (características del usuario).
- Los datos de uso que reúne los datos de la interacción del usuario con el sistema.
- Los datos del entorno en el que se engloban todos los datos del entorno del usuario diferentes a sus características personales.

Este modelo de usuario es según Vélez y Fabregat (Vélez & Fabregat, 2007) un modelo integral a tener en cuenta en la creación de entornos de aprendizaje más eficientes y satisfactorios para el usuario. La consecuente personalización del aprendizaje ofrecida en base a características del estudiante, como nivel de conocimientos o estilos de aprendizaje o intereses del usuario, ha consistido tradicionalmente en adaptar los contenidos o la navegación.

### **1.1.3 – Sistemas de gestión del aprendizaje y Sistemas de Aprendizaje Adaptativo**

Una de las líneas de desarrollo de la tecnología e-learning, iniciada hace al menos veinte años, consiste en la creación y mejora de sistemas hipermedia adaptativos (en adelante AHS, acrónimo de Adaptive Hypermedia Systems). Estos sistemas ofrecen contenidos a sus usuarios de forma personalizada, es decir, están dotados de la capacidad de adaptar los contenidos, o la navegación, a las características de cada usuario. La aplicación más importante de los AHS se refiere al aprendizaje adaptativo, los sistemas que lo soportan son llamados Sistemas de Aprendizaje Adaptativo. El objetivo de tales sistemas es incorporar la tutoría personal de la enseñanza presencial al aprendizaje electrónico (Grimón, Guevara, & Ma Monguet, 2010).

Por otro lado, los sistemas de gestión del aprendizaje (en adelante, LMS, acrónimo de Learning Management Systems) se han desarrollado de tal manera que ofrecen entornos de aprendizaje con un amplio abanico de herramientas de aprendizaje, con facilidades para la gestión y seguimiento de usuarios y para la administración de cursos.

Ambas líneas de desarrollo (AHS y LMS) han ido convergiendo, de modo que los LMS más populares tienden a integrar la posibilidad de personalizar el aprendizaje. La plataforma de e-learning Moodle (acrónimo de Modular Object-Oriented Dynamic Learning Environment), de código abierto, no ha sido ajena a esa idea (Castelló, Lerís, Martínez, & SeinEchalu, 2010).

## **1.2 - Plataforma educativa Moodle**

Moodle es una aplicación web de tipo Ambiente Educativo Virtual de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. La versión más reciente es la 3.1+. Fue creado por Martin Dougiamas, quien fue administrador de WebCT en la Universidad Tecnológica de Curtin. Basó su diseño en las ideas del constructivismo en pedagogía que afirman que el conocimiento se construye en la mente del estudiante y en el aprendizaje cooperativo. La primera versión de la herramienta apareció el 20 de agosto de 2002, a partir de allí han aparecido nuevas versiones de forma regular.(Moodle, 2015).

Moodle como la mayoría de las plataformas educativas modernas no solo gestiona el aprendizaje sino también el contenido de los cursos contenidos en ella, de ahí que se combinen los acrónimos LMS y CMS (Content Management System) para conformar los LCMS (Learning Content

Management System). Para esto soporta disímiles contenidos como textos, animaciones, videos e imágenes que, relacionados entre sí permiten que los estudiantes tengan un curso atractivo.

Este LCMS posee una estructura modular, cuenta con recursos, actividades, bloques, filtros y además sus funcionalidades pueden ser extendidas mediante el desarrollo o utilización de plugins.

Los módulos son componentes auto controlados que extienden las funcionalidades de una aplicación. Los recursos son definidos como mecanismos a utilizar para apoyar el aprendizaje, presentándoles la información a los usuarios. Las actividades llevan la labor práctica del curso. Constituyen funciones que se desarrollan para que los estudiantes interactúen entre ellos y con el profesor (Aguiar, 2013).

Los filtros permiten la transformación automática del texto introducido en resultados, a menudo más complejos. Pueden usarse para añadir de manera automática enlaces, convertir expresiones matemáticas o emoticones a imágenes e insertar reproductores multimedia (Paisant, 2011).

### **1.2.1 – Arquitectura**

Moodle es un sistema que ha sido diseñado de acuerdo con los siguientes criterios (Moodle, 2016):

- Debe poder ejecutarse en la más amplia variedad de plataformas.
- Debe ser fácil de instalar, aprender y modificar.
- Debe ser fácil de actualizar desde una versión a la siguiente.
- Debe ser modular para permitir el crecimiento.
- Debe poder usarse junto a otros sistemas.

Su esencia está en el uso de componentes, para esto cuenta con un núcleo (conocido como “Moodle core”) del sistema rodeado de una gran cantidad de módulos que proveen funcionalidades específicas a la plataforma. La estructura de los módulos en Moodle es muy simple, compuesta por un directorio con archivos PHP, JavaScript, XML y/o CSS y puntos de entrada definidos en varios archivos clave como `locallib.php`, `db/install.xml` o `lib.php` que permiten la comunicación con el núcleo.

### **El Núcleo de Moodle (Moodle Core)**

El Núcleo de Moodle proporciona todos los mecanismos necesarios para el funcionamiento de la plataforma. Implementa todos aquellos mecanismos que luego los módulos utilizan para desarrollar su función. Ofrece un acceso seguro a las bases de datos, encapsula la presentación y su API es extensa y bien documentada. Es por lo tanto la parte más importante del sistema Moodle y sus componentes principales son los siguientes: (Aguiar, 2013)

#### **1.2.2 – Extensiones de Moodle (Plugins)**

Un plugin o una extensión es una aplicación informática que añade funcionalidades específicas a un programa principal. Su nombre procede del inglés (plugin significa “enchufable”) y su presencia es muy habitual en los navegadores web, en reproductores de música y en sistemas de gestión de contenidos (Aguiar, 2013).

Las extensiones en Moodle pueden ser de varios tipos: bloques, matriculación, temas y actividades. Algunos de los plugins más importantes en Moodle son:

- **Actividades y Recursos:** componen los objetos individuales del curso. Existen actividades variadas como: colaborativas, foro, wiki, y cuestionarios. Es el módulo más desarrollado y se instala en el directorio mod dentro de la raíz de Moodle.
- **Bloques:** los bloques son cajas con contenido añadido en columnas a los lados del contenido principal que ofrecen varias funcionalidades. Residen en el directorio blocks.
- **Matriculación:** estos tipos de plugin no son muy comunes en la plataforma, los mismos residen en el directorio enrol.
- **Temas:** los estilos se desarrollan dentro de módulos que existen dentro de la plataforma que residen en el directorio themes (Curbelo, 2013).

El plugin cumplirá con una estructura definida por los desarrolladores de la plataforma; para garantizar el buen funcionamiento del mismo. A continuación, se describe esta estructura y en qué consiste cada uno de sus elementos.

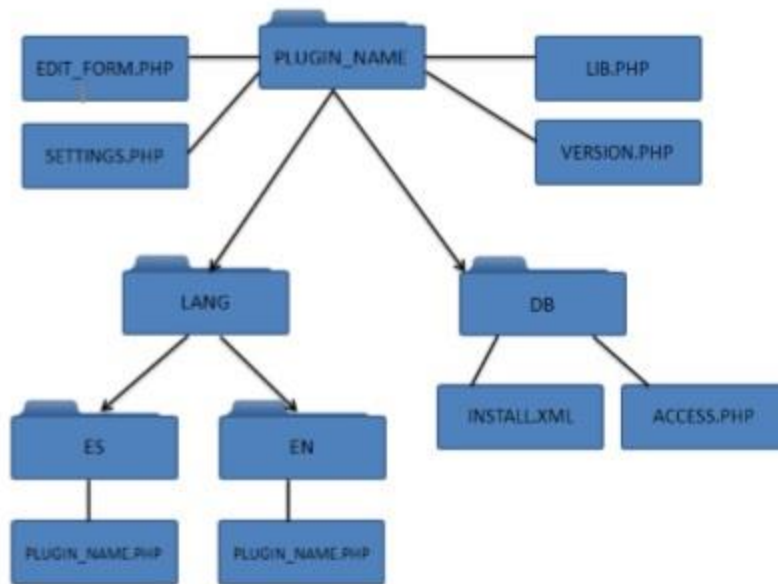


Fig. 2 – Estructura de un plugin

- Plugin\_Name: carpeta principal que incluye todos los componentes del plugin, debe tener el nombre del mismo. Esta carpeta incluye las subcarpetas Lang y DB.
- Lang: carpeta que incluye los idiomas para la internacionalización del plugin.
- DB: incluye el componente install.php y access.php; estos permiten configurar las tablas que se desean agregar a la base de datos, así como los permisos necesarios.
- Edit\_Form.php: se utiliza para configurar cada una de las instancias del plugin.
- Settings.php: se utiliza para la configuración general del plugin.
- Lib.php: incluye todas las funcionalidades, desde la creación de la instancia del plugin, hasta la destrucción del mismo.
- Version.php: especifica la versión del plugin y para qué versiones de Moodle fue desarrollado.
- Install.xml: en este componente se especifica cada una de las tablas que se desean agregar a la base datos y que son necesarias para el correcto funcionamiento del plugin.
- Access.php: componente que contiene los permisos asociados al plugin.
- Plugin\_Name.php: componente que contiene la información que se debe mostrar en la interfaz, dependiendo del idioma que se escoja y que esté configurado.



### **1.2.3 – Rutas de aprendizaje en Moodle**

Las Rutas del Aprendizaje son orientaciones pedagógicas y didácticas para una enseñanza efectiva de las competencias de cada área curricular. Ponen en manos de los docentes, pautas útiles para los diferentes niveles educativos. Estas orientaciones van encaminadas a diferenciar los contenidos para cada nivel de competencia del estudiante (Maguiña & otros, 2015).

Se puede decir que las rutas de aprendizaje pretenden dar atención a las diferencias individuales de los estudiantes desde el propio diseño de la asignatura, esto traducido a los LMS implica mejores orientaciones, variedad de recursos y sistemas de actividades y evaluaciones pensadas para que el estudiante avance según los conocimientos que adquiere.

Moodle permite crear rutas de aprendizaje en sus cursos, a través de las llamadas actividades condicionales o restricciones de acceso. Estas no son más que un conjunto de reglas que permiten acceder a dichas actividades solo si el alumno cumple con determinadas condiciones impuestas por el profesor. Las actividades condicionales son una característica que le permite a un profesor el liberar Recursos y Actividades a sus estudiantes basándose en ciertos criterios tales como la fecha, la calificación recibida o los niveles de participación.

Estas están presentes en todas las actividades de Moodle e incluso secciones enteras de un curso pueden ser configuradas para que el alumno solo las acceda si cumple con ciertas condiciones. Estas opciones incluidas en Moodle solo definen lo que el profesor en su diseño del curso pueda ser capaz de prever para sus estudiantes y aunque es un paso de avance no tiene en cuenta otras posibilidades de individualización.

### **1.3 – Estudio de sistemas similares**

En este campo, solo los sistemas de hipermedia adaptativa utilizan técnicas de Inteligencia Artificial para intentar conocer el estilo de aprendizaje del estudiante. La mayoría de estos sistemas inician sometiendo a los estudiantes a un cuestionario con un grupo de preguntas encaminadas a determinar su estilo de aprendizaje. Entre estos cuestionarios el conocido como Test de Felder-Silverman es el más destacado por su completitud y resultados (Rojas & Alfonso, 2015).

A pesar que estos cuestionarios son la manera más fiable para determinar el estilo de aprendizaje los estudiantes los encuentran tediosos, el test de Felder-Silverman tiene 44 preguntas, y más si tienen que realizarlo para cada curso al que se enfrentan dentro del sistema.

A partir del estudio de algunos de los sistemas de hipermedia adaptativa y LMS más populares se puede concluir que para personalizar el aprendizaje es necesario que el sistema determine el estilo de aprendizaje de los estudiantes, logre hacer seguimiento del mismo por períodos de tiempo prolongado, sea capaz de modificar el estilo si fuera necesario y sería deseable que el profesor pudiera aplicar esta adaptación a todos o solo a algunos (Peña, Marzo, Rosa, & ., 2012).

A continuación, se muestran los sistemas más representativos que realizan adaptación de contenido o adaptación del aprendizaje haciendo énfasis en sus características distintivas.

Características	Sistemas				
	INSPIRE	MASPLANG	CoMoLE	TANGOW	Moodle
Test Inicio	Si	Si	Si	Si	No
Variabilidad de estilo	No	No	No	No	No
Seguimiento global	No	No	No	No	No
Aplicación individual	No	No	Sí	No	No
Interacción profesor - diseño	No	No	No	No	No
Actividades condicionales	No	No	No	No	Sí

Tabla 1: Sistemas de Hipermedia adaptativa y LMS

También se pueden destacar algunas características relevantes y desventajas de estos sistemas:

Sistemas	Característica relevante	Desventaja
INSPIRE	Propone un recorrido de acuerdo al progreso del estudiante	No presenta mecanismo de retroalimentación del estilo de aprendizaje
MASPLANG	El mecanismo de IA implementado es Sistema Multiagente	Orientada por contenidos. No ofrece ejercicios para comprobar el grado de conocimiento adquirido
CoMoLE	Separación de estructuras, materiales y fragmentos	Curso particular
TANGOW	Separación de estructuras, materiales y fragmentos	Necesita información previa de un número importante de usuarios
Moodle	Posibilita crear y compartir cursos y/o recursos. Posibilita crear cursos de manera colaborativa	No adapta la presentación de contenido

Tabla 2: Características y ventajas de los sistemas.

Como se puede observar en los datos en la tabla 2, ningún sistema, una vez que establece un estilo para un estudiante es capaz de modificarlo. Además, ninguno trabaja con datos globales del sistema que le permitan seguir las variaciones de estilo en un tiempo prolongado de uso de la plataforma por parte del estudiante. Excepto CoMoLE ninguno ofrece la posibilidad de que el profesor decida a qué estudiantes realizar la adaptación del aprendizaje. Y solo Moodle y CoMoLE son capaces de permitir actividades condicionales para que el profesor en el diseño del curso pueda trazar ciertas rutas de aprendizaje.

Es apreciable que ninguno de los sistemas analizados cumple con todas las características que plantea Clara Inés Peña que debe poseer un sistema que adapta el aprendizaje.

Además, se puede concluir que, algunos de ellos no realizan técnicas de adaptación sobre el diseño de aprendizaje del curso lo cual es muy importante para poder asignar un contenido al estudiante y varios de ellos determinan el estilo de aprendizaje, pero no hacen uso del mismo para poder realizar un tratamiento diferenciado al estudiante. Moodle no adapta la presentación de contenido, aunque algunos efectos parecidos pueden lograrse utilizando las restricciones de acceso. Ninguno de ellos retroalimenta al profesor en cuanto a los estilos de aprendizaje de sus alumnos.

#### ***1.4 – Alternativas a las técnicas clásicas de recomendación de contenidos.***

Existen diferentes técnicas para recomendar contenidos, recursos o productos a los usuarios de los sistemas informáticos. Casi todas manejan de una manera u otra las probabilidades de que un usuario prefiera un producto basado en los conocimientos previos sobre el propio usuario. Cuando no se tiene información de un usuario se utilizan técnicas que van desde aplicar cuestionarios hasta buscar similitudes con otros grupos de usuarios, este problema en los sistemas de recomendación se llama arranque en frío.

Las técnicas (no basadas en inteligencia artificial) que mejor describen las acciones de los usuarios en un sistema están basadas en la teoría de los juegos y tratan de estimar a través de cálculos probabilísticos cuáles serán las futuras acciones del usuario, permitiendo al sistema tomar decisiones. La toma de decisiones comienza con el concepto de preferencia. Una decisión ocurre cuando una persona se enfrenta a varias opciones (alternativas) que tienen distintas

características (criterios), evalúa las consecuencias de elegir cada una de ellas y, finalmente, elige una dependiendo de sus preferencias (Rustichini, Stephen, Carpenter, & Götte, 2008).

Extrapolando estos criterios de la toma de decisiones, aplicadas a los juegos, a las plataformas educativas se puede utilizar estas técnicas para predecir el comportamiento de un estudiante al enfrentarse a ciertos tipos de recursos y analizando la posibilidad de mostrar otros recursos acordes a sus preferencias y estilos de aprendizaje.

### **Teorías representativas de toma de decisiones**

El estudio de las decisiones humanas se ha desarrollado tradicionalmente en el campo de la economía y la teoría de juegos y se centra en comportamientos normativos que definen el hecho de que un sujeto tiene que elegir siempre la alternativa que maximice la recompensa asociada a las consecuencias de una decisión (Iglesias Sánchez , 2013).

Dentro de este campo se han desarrollado diversas teorías entre las que destacan:

- Teoría del Valor Esperado
- Teoría de la Utilidad Esperada
- Teoría de la Perspectiva
- Teoría de la Perspectiva Acumulativa
- Arquitectura de Ho

En este grupo de teorías las más completas son las dos últimas que no solo tienen en cuenta la probabilidad de ocurrencia de un evento sino también el tiempo en que puede ocurrir el mismo.

### **Teoría de la Perspectiva Acumulativa**

La probabilidad de un resultado se calcula mediante las probabilidades de todos los resultados posibles. Si se ordena los resultados de mayor a menor y el valor que divide las ganancias de las pérdidas está en la posición z-ésima, entonces ( Kahneman & Tversky, 1992):

$$S(A_j) = \sum_{i=1}^n v(x_{ij}) \cdot \pi_{ij} \quad (7)$$

donde:

$$\pi_{ij} = \begin{cases} \pi(p_{ij}) & \text{si } i = n \\ \pi(\sum_{k=i}^n p_{kj}) - \pi(\sum_{k=i+1}^n p_{kj}) & \text{si } i \geq z \\ \pi(\sum_{k=0}^i p_{kj}) - \pi(\sum_{k=0}^{i-1} p_{kj}) & \text{si } i < z \\ \pi(p_{ij}) & \text{si } i = 0 \end{cases} \quad (8)$$

La Teoría de la Perspectiva Acumulativa se caracteriza por los parámetros:  $\alpha$ ,  $\beta$ ,  $\lambda$ ,  $\delta$  y  $\gamma$ .

### Arquitectura de Ho

Esta arquitectura sigue los axiomas de la Teoría de la Perspectiva pero usa funciones más sencillas. La arquitectura de Ho distingue entre ganancias y pérdidas, como la Teoría de la Perspectiva, y añade el retardo temporal que existe desde que se toma una decisión hasta que se obtiene el resultado como un factor más en el proceso de la toma de decisiones. De acuerdo a esta arquitectura, las alternativas se puntúan mediante la siguiente ecuación (Ho, Mobini , Chiang , Bradshaw , & Szabadi, 1999):

$$S(A_j) = \sum_{i=1}^n \begin{cases} f(x_{ij}, Q) \cdot g(d_{ij}, K) \cdot h(p_{ij}, H) & \text{si } x_{ij} \geq 0 \\ -f(-x_{ij}, Q') \cdot g(d_{ij}, K') \cdot h(p_{ij}, H') & \text{si } x_{ij} < 0 \end{cases} \quad (10)$$

La siguiente expresión describe la función de valor ( $f(\cdot)$ ) utilizada en la arquitectura de Ho.

$$f(x, Q) = \frac{x}{x + Q} \quad (11)$$

La expresión sólo depende de un parámetro, que es diferente para las ganancias ( $Q$ ) y para las pérdidas ( $Q'$ ). Estos dos parámetros determinan la curva de la función de valor en ambos casos.

La función  $g(\cdot)$  es decreciente y define un descuento sobre los resultados que se obtienen con alguna demora en el tiempo (Chapman, 2005). Los seres humanos tienden a reducir el valor subjetivo de una recompensa cuanto mayor sea el retardo ( $d$ ) para obtenerla. El parámetro  $K$  determina el grado de descuento temporal (Chapman, 2005).

$$g(d, K) = \frac{1}{1 + K \cdot d} \quad (12)$$

La función de probabilidad ( $h(\cdot)$ ) es la siguiente:

$$h(p, H) = \frac{p}{p + H \cdot (1 - p)} \quad (13)$$

Al igual que en la función de valor, la función de probabilidad está definida por un parámetro que varía dependiendo de si está representando ganancias ( $H$ ) o pérdidas ( $H'$ ). Estos dos parámetros determinan la curva de la función de probabilidad  $h(\cdot)$  (Iglesias Sánchez , 2013).

Después del análisis de estas teorías se consideró que la Arquitectura de Ho es la más completa y la que más se ajusta para ser utilizada en plataformas educativas ya que sus valores son perfectamente adaptables a la información que se puede obtener sobre los estudiantes de un curso. A pesar de que no se tiene constancia de que se haya utilizado en este campo si es ampliamente utilizada en la toma de decisiones para videojuegos.

## **1.5 – Metodologías**

Para el proceso de desarrollo del software se hace necesario el uso de una metodología que permita llevar a cabo la correcta elaboración del producto. Estas están formadas por fases que guiarán a los desarrolladores de sistemas a seleccionar las técnicas propicias en cada parte del proyecto además de planificarlo, gestionarlo, controlarlo y evaluarlo. Una metodología refiere una representación que facilita la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes implicadas en la construcción del sistema. Tiene como finalidad mostrar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desplegar software de calidad, incluyendo heurísticas de construcción y criterios de comparación de modelos de sistemas. (Jorge & Fuentes, 2015).

Las metodologías de desarrollo de software se dividen en dos grupos: metodologías pesadas o tradicionales y metodologías ágiles.

### **Metodologías Pesadas**

Las metodologías pesadas están fundamentadas en normas procedentes de estándares seguidos por el entorno de desarrollo, brindan por lo general cierta resistencia a los cambios que puedan ocasionarse durante el ciclo de desarrollo del producto, son procesos muy bien

controlados con numerosas políticas y normas a seguir (Pressman, 2010). Su principal exponente es el Proceso Unificado de Rational (RUP por sus siglas en inglés), la misma no se considera un sistema con pasos firmemente establecidos sino un conjunto de técnicas adaptables al contexto de cada organización.

### **Metodologías Ágiles**

Los expertos cuando caracterizan a favor de la metodología ágil emplean la frase: “los métodos tradicionales están atrapados en un pantano y producirán una documentación sin defectos en vez de un sistema funcional que satisfaga las necesidades del negocio” (Pressman, 2010). Las metodologías ágiles se basan en dos aspectos puntuales, el retrasar las decisiones y la planificación adaptativa; permitiendo potenciar el desarrollo de software a gran escala (Letelier, 2008).

Como resultado de esta nueva teoría se crea un Manifiesto Ágil cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Teniendo en cuenta el análisis de los dos tipos de metodologías se llega a la conclusión de que el proceso de desarrollo de software debe ser guiado por una metodología ágil ya que son las adecuadas a las características del proyecto. En la presente investigación se cuenta con poco tiempo de desarrollo, un equipo de desarrollo pequeño y se trabajará junto al cliente.

Dentro de las metodologías ágiles se destacan las metodologías XP (del inglés Extreme Programming, del español Programación Extrema), CRYSTAL, SCRUM y Ágil UP.

En la siguiente tabla se realiza una comparación entre ellas basándose en 3 aspectos fundamentales (Rojas & Alfonso, 2015):

1. Vista del sistema como algo cambiante: capacidad de adaptación a los cambios.

2. Colaboración entre los miembros del equipo: es el nivel de intercambio de información entre los miembros del equipo.
3. Características más específicas de la propia metodología:
  - Simplicidad: es el nivel sencillez de las soluciones implementadas que generan.
  - Excelencia técnica: Define el grado de calidad que presenta.
  - Resultados: utilidad y calidad del producto.
  - Adaptabilidad: es la propiedad que posee de adaptarse a cualquier proyecto.

El ranking de agilidad es de 1 a 5 siendo 1 menos ágil y 5 más ágil (Highsmith, 2002).

**Media de las Características Metodológicas (CM):** sumatoria de todos los valores de CM / cantidad de elementos CM.

**Media Total:**  $SC^2 + C^3 + \text{Media CM}$  / cantidad de elementos.

	AUP-UCI	Crystal	Scrum	XP
<b>Sistema como algo cambiante</b>	2	4	5	<b>5</b>
<b>Colaboración</b>	3	5	5	<b>5</b>
<b>Características Metodología (CM)</b>				
<b>-Resultados</b>	3	5	5	<b>5</b>
<b>-Simplicidad</b>	2	4	5	<b>5</b>
<b>-Adaptabilidad</b>	3	5	4	<b>3</b>
<b>-Excelencia técnica</b>	4	3	3	<b>4</b>
<b>-Prácticas de colaboración</b>	3	5	4	<b>5</b>
<b>Media CM</b>	3.0	4.4	4.2	<b>4.4</b>
<b>Media Total</b>	2.7	4.5	4.7	<b>4.8</b>

Tabla 3: Comparación entre metodologías ágiles.

Como se puede apreciar XP posee los valores más altos por tanto es la que mayor grado de agilidad ofrece.

Según el reporte “The 10 annual state of agile report” del 2015 que realiza anualmente la compañía VersionOne dedicada al análisis de los temas, la metodología ágil más utilizada en el mundo es XP seguida de Scrum híbrido con XP y AUP. Esto se debe principalmente a que XP

<sup>2</sup>SC: Sistema como algo cambiante

<sup>3</sup>C: Colaboración



considera que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos (Versione, 2015). Esta investigación marcada por algunos aspectos empíricos requerirá de muchos cambios y ajustes en los requisitos haciendo a esta metodología ideal para el desarrollo sobre otras metodologías ágiles.

### **Programación Extrema (XP)**

La Programación Extrema es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. Este tipo de método se basa en una realimentación continuada entre el cliente y el equipo de desarrollo con una comunicación fluida entre todos los participantes, también busca simplificar las soluciones implementadas y coraje para los múltiples cambios. Este tipo de programación es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo (Curbelo, 2013).

Si bien el ciclo de vida de un proyecto XP es muy dinámico, se puede separar en fases (Pressman, 2010). Estas se describen a continuación.

**Fase de exploración:** en esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información. El resultado es una visión general del sistema y un plazo total estimado.

**Fase de planificación:** la planificación es una fase corta, en la que el cliente y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, y las entregas.

**Fase de iteraciones:** esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración.

**Fase de puesta en producción:** si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa.

## **1.6 – Selección de los lenguajes, herramientas y tecnologías a utilizar**

Moodle está programado en PHP, haciendo un uso intensivo de los archivos XML para la descripción de los componentes y la estructura de las tablas de la base de datos. Cualquier nueva funcionalidad que desee aprovechar las características de Moodle e interactuar con su núcleo debe ser escrita utilizando estas tecnologías. Además del lado del cliente Moodle utiliza CSS y JavaScript para dotar a sus páginas de una presentación de calidad y una mejor interactividad.

Estas características son las que deciden que para el desarrollo de la solución del problema identificado en la presente investigación sean seleccionadas dichas tecnologías.

### **1.5.1 –Tecnologías para el desarrollo web para el lado cliente**

#### **Lenguaje Marcado de Hipertexto 5.0:**

El Lenguaje Marcado de Hipertexto (HTML) es el predominante para elaborar las páginas web. Mediante HTML se pueden estructurar las mismas usando etiquetas predefinidas que marcan el inicio y el final de cada elemento. Permite insertar hipertextos (enlaces) que posibilitan la comunicación de estas con otras páginas.

Actualmente, HTML se usa para estructurar y proporcionar una descripción semántica a las páginas web, que son las funciones para las que se creó en un principio. Por tanto, no determina ni la apariencia ni el comportamiento interactivo de las mismas, sino su estructura y el significado de sus elementos (Aguiar, 2013).

#### **Hojas de Estilo en Cascada 3.0:**

El estándar Hojas de Estilo en Cascada (CSS) se utiliza para la aplicación de estilos que definen los aspectos de todos los contenidos, como es el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, y la tabulación con la que se muestran los elementos de una lista. Es un mecanismo simple que permite definir una serie de estilos para luego aplicarlos a una página, e incluso, a un grupo determinado. Su objetivo es separar la estructura de un documento de su presentación, haciéndose imprescindible para crear páginas web complejas (Aguiar, 2013).

Una característica esencial de este lenguaje es que permite separar los elementos formales de los de contenido. De esta forma, solo con una pequeña modificación de la hoja de estilo es posible cambiar el aspecto de todas las páginas en las que se aplica esa hoja. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre el estilo y formato de sus documentos (Aguiar, 2013).

### **Lenguaje Marcas Extensible 1.0:**

El Lenguaje de Marcas Extensible (XML) es un estándar que permite el intercambio de información estructurada entre diferentes plataformas. Por este motivo es una forma universal para documentos y datos estructurados en Internet, y se puede usar en bases de datos, editores de texto y hojas de cálculo. Admite compartir la información de una manera segura, fiable y fácil. Facilita el intercambio de una amplia variedad de datos en la web. Está diseñado para hacer frente a los problemas de compatibilidad y adaptabilidad de las nuevas tecnologías en Internet (Curbelo, 2013).

### **JavaScript 1.2:**

Es un lenguaje que permite crear páginas que posibilitan el intercambio con los usuarios. Entre las características más relevantes se encuentran: es un lenguaje basado en acciones que poseen menos restricciones. Gran parte de su programación está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas, entre otros (Curbelo, 2013).

### **Framework de desarrollo**

#### **jQuery 1.12.0:**

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos y desarrollar animaciones a páginas web.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados (jQuery, 2016). Desde la versión 2.2 de Moodle esta plataforma incorpora jQuery como framework

para el trabajo del lado del cliente.

### **Angular 1.3.3:**

AngularJS es un proyecto de código abierto, realizado en Javascript que contiene un conjunto de bibliotecas útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. Además implementa el patrón arquitectónico Modelo-Vista-Controlador el cual permite repartir la lógica de la aplicación por capas, lo que resulta muy adecuado para aplicaciones de negocio ( Basalo & Alvarez, 2014).

## **1.5.2 – Tecnología de desarrollo web para el lado servidor**

### **Procesador de Páginas de Hipertexto 5.4.4:**

Procesador de Páginas de Hipertexto (PHP por sus siglas en inglés Preprocessed Hypertext Pages) es un lenguaje interpretado, diseñado para desarrollar páginas web dinámicas de forma rápida, el cual puede ser impregnado en páginas HTML. Es de alto nivel y se muestra como una alternativa de fácil acceso por su naturaleza de código abierto (Aguiar, 2013).

En los requerimientos para la versión 2.8 de Moodle se necesita como mínimo PHP 5.4.4.

### **PostgreSQL 9.1:**

PostgreSQL es un potente sistema de gestión de bases de datos objeto-relacional (O- RDBMS), multiusuario, centralizado y de propósito general, que está siendo desarrollado desde 1977 y está liberado bajo la licencia Berkeley Software Distribution (BSD). Es ampliamente considerado como el sistema gestor de base de datos de código abierto más avanzado del mundo (postgresql, 2016).

Moodle puede trabajar con distintos gestores de Bases de Datos, pero para el desarrollo de la solución se utiliza PostgreSQL por ser el que está desplegado en el Nodo de la UCI.

## **1.5.3 – Servidor web**

Un servidor web es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web

utilizan el protocolo HTTP. Los servidores web están alojados en un ordenador que cuenta con conexión a Internet. El servidor web, se encuentra a la espera de que algún navegador le haga alguna petición, como por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red (apache, 2016).

#### **Apache 2.4.9:**

Apache es un poderoso servidor web, cuyo nombre proviene de la frase inglesa “a patchy server” y es libre, de código abierto y se distribuye bajo licencia GPL. Una de las ventajas más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede trabajar con diferentes sistemas operativos y mantener su excelente rendimiento (apache, 2016).

En el caso de esta investigación ya la plataforma Moodle se encuentra corriendo sobre un servidor Apache en su versión 2.4.9 desplegada en el nodo de la UCI.

#### **1.5.4 – Herramienta CASE**

La herramienta CASE a utilizar será Visual Paradigm. La misma está diseñada para una amplia gama de usuarios, incluyendo Ingenieros de Software, Analistas de Sistemas, Analistas de Negocios y Arquitectos de Sistemas que estén interesados en la creación de grandes sistemas de software de manera confiable, a través del paradigma Orientado a Objetos. Visual Paradigm- Unified Modeling Language (VP-UML) soporta los últimos estándares de anotaciones de JAVA y UML, y provee soporte para la generación de código y la ingeniería inversa para Java.

Las transiciones del análisis al diseño, y de este a implementación están adecuadamente integradas dentro de la herramienta CASE, de manera que reduce significativamente los esfuerzos de todas las etapas del ciclo de desarrollo de software (Curbelo, 2013).

#### **1.5.5 – Entorno de Desarrollo Integrado**

Un entorno de desarrollo integrado (IDE), es un programa informático compuesto por un conjunto de herramientas de programación que puede ser dedicado a un lenguaje o para varios. La mayoría de los IDE tienen autocompletado inteligente de código (IntelliSense). Algunos IDE contienen un compilador, un intérprete, o ambos, tales como NetBeans y Eclipse.

#### **Net Beans v8.0:**

NetBeans es un reconocido entorno de desarrollo integrado de código abierto y de distribución gratuita apoyado por una amplia comunidad de desarrolladores. Entre sus funciones incluye resaltado de sintaxis para los lenguajes de programación PHP, HTML, JavaScript, CSS, entre otros. Brinda depuración y un potente completamiento de código. Integra la gestión de configuración y las herramientas de control de versiones, lo que contribuye a mejorar el trabajo en equipo (NetBeans, 2016).

## **1.7 – Conclusiones**

En el estudio desarrollado en este capítulo se han abordado los principales aspectos de interés para el desarrollo de esta investigación permitiendo a los autores arribar a las siguientes conclusiones parciales:

- La atención a las diferencias individuales en la educación a distancia cobra vital importancia, de ahí que aprovechar los estilos de aprendizaje de cada usuario para proponer qué recursos estudiar es una opción viable para ganar la atención del usuario.
- Ninguno de los sistemas estudiados permite adaptar su actuación al trabajo de la plataforma Moodle utilizada en la Universidad de las Ciencias Informáticas, por lo que se hace necesario entonces desarrollar un sistema que personalice el aprendizaje para cursos basados en esta plataforma.
- La utilización de teorías de tomas de decisiones ofrece una alternativa probabilística que permite decidir o prever los gustos o preferencias del usuario, remarcando así su estilo de aprendizaje en función de los recursos educativos en la plataforma.
- La arquitectura modular de Moodle favorece la incorporación de estas teorías y funciones a través del desarrollo de plugins.

## **Capítulo 2: Propuesta de solución**

### ***Introducción***

En el presente capítulo se abordarán temas fundamentales para el desarrollo del componente que sea capaz de satisfacer al cliente. Se hará alusión a la fase Exploración y Planificación de la metodología XP. Se elaborará el levantamiento de la lista de reserva, se generarán las historias de usuario correspondientes a la misma, el plan de iteraciones y el plan de entrega los cuales tienen un impacto significativo en la caracterización y el esclarecimiento del componente que se desea implementar.

### ***2.1 – Descripción de la solución***

La solución propuesta consiste en un módulo de Moodle que muestre al estudiante de un curso los recursos y actividades que le resulten más adecuados para aprender en función de su estilo de aprendizaje. La determinación del estilo de aprendizaje se hará a partir de la experiencia de usuario en el sistema evitando que este tenga que realizar un tedioso cuestionario antes de cada curso.

Al diseñar un curso, el profesor o el responsable del diseño instruccional, debe tener en cuenta la creación de recursos que cubran los cuatro estilos de aprendizaje. En la configuración del módulo el profesor asignará los recursos a uno de los estilos de aprendizaje confeccionando de esta forma las rutas de aprendizaje predeterminadas para cada estilo. Cuando se registra un nuevo estudiante en el curso, el módulo buscará si ya este tiene asociada una experiencia de usuario a partir de cursos anteriores. De no ser así realizará una consulta a la base de datos de Moodle para determinar si el usuario ha participado en cursos previos y con esa información calcular una experiencia de usuario. Si no existiera ninguna información el estudiante iniciará el curso en una ruta por defecto, hasta la primera evaluación.

Cuando se llega a la primera evaluación se hace un cómputo de la interacción del estudiante con el curso y se organiza el mismo en función del estilo de aprendizaje, mostrándole los recursos y actividades asociadas a este.

El profesor puede interactuar con el módulo para obtener los reportes del curso en cualquier momento del mismo. Podrá obtener datos como los recursos y actividades más utilizados, la experiencia de usuario para cada participante y con esa información puede arribar a conclusiones relacionadas con el proceso de aprendizaje. Al terminar el curso será posible seleccionar como ruta predeterminada la más usada por los estudiantes para futuras ediciones.

Para un mejor entendimiento de los conceptos asociados al negocio y sus relaciones se puede consultar el siguiente modelo conceptual.

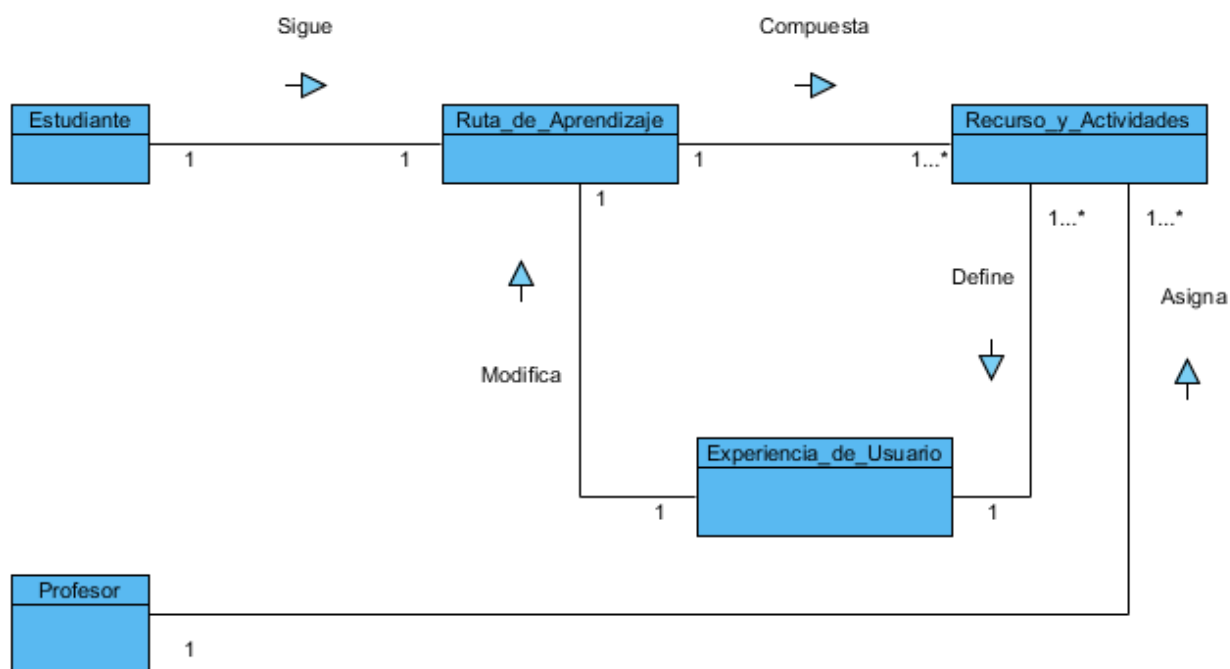


Fig. 3 – Modelo Conceptual

**Ruta de aprendizaje:** marca cuáles recursos se muestran a los estudiantes en función de sus estilos de aprendizaje. Según se avanza en el curso se reevalúan los estilos por lo que los estudiantes podrían cambiar de ruta.

**Recursos y actividades:** es todo aquello que el profesor expone en el curso para que los estudiantes aprendan e interactúen con el sistema.



**Experiencia de usuario:** se determina en función de las características del modelo de usuario para a partir de la misma determinar un estilo de aprendizaje. Incluye las preferencias del usuario, la interacción con la plataforma y los posibles datos del entorno que se puedan conseguir.

Como se puede observar la adaptación del aprendizaje y la decisión de los estilos se hace a partir del cálculo de la experiencia de usuario, la cual está altamente influenciada por los tipos de actividades. Las actividades llevan una ponderación o peso que determina el profesor en función de la importancia de las mismas dentro del diseño del curso.

### **2.1.1 – Tipos de Actividades y Ponderaciones.**

Una actividad es un nombre general para un grupo de características en un curso Moodle. Usualmente una actividad es algo que un estudiante hará, y donde interactúa con otros estudiantes, con el maestro o el propio sistema. Dentro de Moodle estas actividades pueden clasificarse en cuatro grupos o tipologías (Moodle, 2016):

- Individuales (tareas, base de datos, examen, foro, glosario, juego, lección, scorm, taller, wiki, IMS)
- Colaborativas (base de datos, encuestas, foro, glosario, taller, wiki)
- Comunicación (foro)
- Contenidos (archivos, carpetas, etiquetas, libro, página, IMS, URL, herramienta externa)

Para los efectos de un sistema de adaptación del aprendizaje es necesario realizar ponderaciones que permitan evaluar el rendimiento del estudiante y su interacción con la plataforma. Los niveles de importancia de las actividades para poder realizar las ponderaciones mostradas en este trabajo se determinaron de manera empírica a partir del trabajo realizado por los profesores y especialistas del CENED. Estos valores pueden tomar un rango de uno a tres y pueden ser modificados por el profesor en función de la relevancia prevista para las actividades durante el diseño del curso. Las actividades con nivel de importancia uno son consideradas las más importantes y disminuye el nivel a medida que el valor aumenta. Los valores por defecto se muestran en la siguiente tabla.

Actividades	Nivel de importancia
tareas, examen, foro, lección	1
base de datos, glosario, scorm, taller, wiki, libro, IMS	2
encuestas, herramienta externa, juego, archivos, carpetas, etiquetas, página, URL, foro	3

Tabla 4: Nivel de importancia de las actividades.

Otro aspecto que influye en la interacción con la plataforma es el tiempo de actividad (TA) que no es más que el tiempo que un estudiante se toma en realizar una actividad. Este factor es de vital importancia ya que es un medidor de la atención que el estudiante presta en la plataforma a los recursos de aprendizaje. Este parámetro trata de deducir un tiempo estimado donde el usuario realmente haya interactuado de manera consciente con la actividad diseñada por el profesor en la plataforma.

Esta variable puede asumir valores de 0 a 1 en función de la siguiente expresión:

Si  $TA < 1$  min entonces  $TA = 0$ , si  $1\text{min} < TA < \text{Tiempo de interacción}$  entonces  $TA = 1$ , si  $TA > \text{Tiempo de interacción}$  entonces  $TA = 0$ .

El tiempo de interacción es un valor que se fija en 15 minutos, si el estudiante permanece más de este tiempo en la misma actividad sin escoger ninguna opción o cambiar de actividad se asume que no está haciendo nada, por lo que el TA empezaría a medirse nuevamente a partir del próximo evento con la plataforma.

### 2.1.2 – Determinación de la experiencia de usuario

Para determinar la experiencia de usuario se tienen en cuenta las variables mencionadas en el epígrafe anterior determinándose valores de Tiempo de Actividad y nivel de importancia para cada actividad realizada por el usuario hasta el momento.

La expresión para determinar la experiencia de usuario está compuesta por la media de las acciones que el estudiante realiza por cada tipología de actividad. Quedando esta expresión como se muestra a continuación.

$$\begin{aligned}
 ExperienciaUsuario = & \sum_{i=0}^n \frac{tareas_i}{n} + \sum_{i=0}^n \frac{Examen_i}{n} + \sum_{i=0}^n \frac{taller_i}{n} + \sum_{i=0}^n \frac{foro_i}{n} + \sum_{i=0}^n \frac{leccion_i}{n} + \\
 & \sum_{i=0}^n \frac{BD_i}{n} + \sum_{i=0}^n \frac{glosario_i}{n} + \sum_{i=0}^n \frac{SCORM_i}{n} + \sum_{i=0}^n \frac{IMS_i}{n} + \sum_{i=0}^n \frac{wiki_i}{n} + \sum_{i=0}^n \frac{recurso_i}{n} + \sum_{i=0}^n \frac{encuesta_i}{n} + \\
 & \sum_{i=0}^n \frac{juego_i}{n}
 \end{aligned} \tag{14}$$

La determinación de las expresiones incluidas en la sumatoria es altamente dependiente del tipo de actividad y el nivel de importancia que se haya definido para cada una. A continuación, se describe el cálculo de cada una de estas expresiones.

Tareas (Assigments):

$$tarea = Tiempo Entrega * Evaluación - (Intentos - 1) \tag{15}$$

Donde:

**Tiempo Entrega** es un elemento binario que asume 1 si el estudiante hizo la tarea y 0 si no la hizo.

**Evaluación** es la nota obtenida en la tarea y normalizada a una escala de 5 puntos

**Intentos** es la cantidad de veces que realizó la tarea si lo tuviera permitido.

Examen, Taller:

$$Examen, taller = Evaluación \tag{16}$$

Foro: Cuando son evaluativos se toma el resultado de la evaluación al igual que en el Examen o Taller, en caso de no ser evaluativos se determina por la siguiente expresión.

$$Foro = \frac{Hilos creados por el usuario}{Hilos del foro} + \frac{replicas del usuario + Tiempo Actividad}{replicas del foro} \tag{17}$$

Donde:

**Hilos creados por el usuario** es la cantidad de hilos de debate que el usuario crea en un foro si tiene permitido hacerlo.

**Hilos del Foro** es el total de hilos de debate existentes en el foro.

**Réplicas del usuario** es la cantidad de respuestas que el estudiante da en el foro como medida de su participación (no tiene en cuenta la calidad de las respuestas).

**Réplicas del foro** es la cantidad total de réplicas en el foro.

Lección: Cuando son evaluativos se toma el resultado de la evaluación sino se toma el promedio de visitas a la lección normalizado a una escala de 5 puntos.

$$\text{Lección} = \text{VisitaPromedio} * TA \quad (18)$$

Donde:

**VisitaPromedio** es el promedio de visitas del estudiante a la lección.

Base de Datos, Glosario: Cuando estas actividades son evaluativas se toma el resultado de la evaluación, en caso contrario se toma el número de entradas a la base de datos normalizado a una escala de 5 puntos.

$$\text{Base de Datos, Glosario} = \text{numero de entradas} * TA \quad (19)$$

SCORM, IMS, Libro:

$$\text{SCORM, IMS, Libro} = \text{VisitaPromedio} * TA \quad (20)$$

Wiki:

$$\text{Wiki} = \text{actualizacionesPromedio} * TA \quad (21)$$

Donde:

**actualizacionesPromedio** es el promedio de actualizaciones que hace el estudiante en la wiki.

Encuesta, Juego:

$$\text{Encuesta, Juego} = \text{Realizada} * TA \quad (22)$$

Donde:

**Realizada** es un término binario que dice si se ha realizado la encuesta o no, para una encuesta efectuada, Realizada = 1; Realizada = 0 en caso contrario.

### 2.1.3 – Determinación del estilo de aprendizaje:

La determinación del estilo de aprendizaje se realiza a partir de la probabilidad de que el estudiante cambie su estilo en función de los recursos y actividades que utiliza frecuentemente en la plataforma expresada en su experiencia de usuario.

Para cada estilo de aprendizaje se determina la probabilidad que tiene de ser el adecuado para el estudiante en función de su interacción con la plataforma. La probabilidad de ocurrencia de determinado estilo está dada por la siguiente expresión:

$$S(A_j) = \sum_{i=0}^n f(x_{ij}, Q) * ExperienciaUsuario() * P(A/B) \quad \text{si } x_{ij} \geq 3,$$
$$\sum_{i=0}^n -f(x_{ij}, Q') * ExperienciaUsuario() * P(A/B) \quad \text{si } 0 \leq x_{ij} < 3 \quad (23)$$

Donde:

**P(A)** – Probabilidad de aprobar.

**P(B)** – Probabilidad de que apruebe con recursos de un estilo dado.

**P(A/B)** =  $\frac{P(A|B)*P(A)}{P(B)}$  que es la probabilidad de que los recursos consultados sean los adecuados para el estudiante.

Función de valor  $f(x_{ij}, Q) = \frac{x_{ij}}{x_{ij}-Q}$  determina la probabilidad de que con esos recursos obtenga una nota  $x_{ij}$ .

La expresión para determinar el estilo de aprendizaje  $S(A_j)$  se calcula 4 veces, una para cada estilo de aprendizaje. Tomando la que exprese la mayor probabilidad y modificando la ruta que lleva el estudiante si fuera necesario.

## **2.2 – Fase de exploración**

Esta es la primera fase de la metodología XP, en la cual, los clientes plantean a grandes rasgos las funcionalidades que son de interés para la elaboración del producto, transformándose las mismas en historias de usuario. Partiendo de la información obtenida, el equipo de desarrollo evaluará de forma general el tiempo de codificación, se familiarizará con las herramientas, tecnologías y prácticas que serán utilizadas en el proyecto, además, se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo para ello.

La fase de exploración toma de pocas semanas a pocos meses, dependiendo de la habilidad que tengan los programadores con las tecnologías seleccionadas. Es válido aclarar que las estimaciones realizadas en esta etapa son primarias, pues estas se basan en datos de alto nivel, los cuales pueden variar a medida que se analicen con mayor cuidado y detalle en las siguientes fases (Jorge & Fuentes, 2015).

## **2.3– Personal relacionado con el sistema.**

Uno de los elementos más importantes que se deben considerar cuando se comienza el desarrollo de un sistema informático es la delimitación de la audiencia a la cual va dirigido el mismo, en algunas ocasiones puede ser un personal relacionado con el sistema o alguien completamente ajeno a él. Debe especificarse que esta audiencia a su vez puede ser dividida en grupos atendiendo a sus necesidades.

Se define como persona relacionada con el sistema, a toda aquella que de una manera u otra interactúe con este, obteniendo un resultado de uno o varios procesos que se ejecutan en el mismo. También son considerados como personas relacionadas con el sistema, aquellas que se encuentran involucradas en dichos procesos, que participan en ellos, pero no obtienen ningún resultado de valor.

<b>Personal</b>	<b>Descripción</b>
Profesor	Tiene completo acceso al módulo y es quien puede decidir qué estudiante usa el módulo y cuál no. Además puede obtener información de él.
Estudiante	Interactúa con los recursos publicados y a través de la experiencia de usuario que se genera se modifica su ruta de aprendizaje.
Administrador	Instala o desinstala el módulo, una vez instalado en el servidor el profesor se encarga de habilitarlo o deshabilitarlo de acuerdo a las necesidades del curso.

Tabla 5: Personal relacionado con el sistema.

### 2.3.1 – Lista de reserva del producto:

La lista de reservas del producto está compuesta por una serie de requisitos funcionales que especifican una acción que debe ser capaz de realizar el sistema. Seguidamente se muestran los requisitos funcionales del módulo (Rumbaugh, 2000).

#### Recursos Funcionales (RF):

1. **Visualizar recursos.**
2. **Visualizar estilos de aprendizaje.**
3. **Asignar estilos a los recursos.**
4. **Visualizar la interfaz de usuario.**
5. **Configurar el módulo.**
  - 5.1 Activar o desactivar módulo.
6. **Generar reporte.**
  - 6.1 Generar reporte de uso de recursos.
  - 6.2 Generar reporte de uso de ruta.

**6.3** Generar reporte de trayectoria de grupo.

**6.4** Generar reporte de uso de estilos de aprendizaje.

**6.5** Generar reporte de experiencia de usuario.

**7. Calcular experiencia de usuario.**

**8. Calcular estilos de aprendizaje.**

**9. Modificar la ruta del estudiante.**

**10. Modificar visibilidad de los recursos.**

**11. Definir el estilo de aprendizaje del estudiante.**

**12. Definir el nivel de importancia de los recursos.**

### **2.3.2 – Aspectos no funcionales del sistema**

Los aspectos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad. (Rumbaugh, 2000)

#### **Software:**

- Moodle debe estar instalado con la versión 2.8, que es la que actualmente se utiliza en la universidad, o superior.
- Versión mínima de PHP: PHP 5.4.4 (aunque siempre se debe utilizar la versión más reciente PHP 5.4.x o 5.5.x).
- Como servidor de base de datos: Postgres teniendo como versión mínima la 9.1.
- Los navegadores que se deben tener instalados son Google Chrome como versión mínima la 30.0, Mozilla Firefox como versión mínima la 25.0, Apple Safari como versión mínima la 6, Microsoft Internet Explore como versión mínima la 9.

#### **Hardware:**

- Espacio de disco: 200 MB para el código de Moodle, más cuanto el profesor necesite para almacenar sus materiales.
- Procesador: 1GHz (mínimo), se recomienda 2GHZ doble núcleo o más.
- Memoria RAM: 512 (mínimo), 1GB o más es fuertemente recomendado.



**Confiabilidad:**

El sistema debe ser preciso con la información que maneja, haciendo énfasis en los resultados de los análisis que ejecutará. Para evitar errores que puedan incidir negativamente en la toma de decisiones.

**Portabilidad:**

El sistema debe ser independiente de plataforma. Debe ejecutarse tanto en Microsoft Windows como en GNU/Linux.

**Apariencia o interfaz externa:**

La interfaz del sistema deberá ser amigable a los usuarios finales. Incluirá solo la información y elementos necesarios que hagan intuitivo su uso. Deberá contar con una adecuada combinación de color y tipografía. En todo momento se debe respetar la tipología y estructura de la interfaz propia de Moodle.

**2.2.3 – Historias de usuario (HU)**

Las HU son comúnmente utilizadas en la metodología XP para especificar con más detalles los requisitos del software tanto funcionales como no funcionales. De la interacción con los usuarios del sistema se elaboraron 12 HU. A continuación, se muestran tres historias de usuarios establecidas para el desarrollo del módulo, el resto se encuentra en los anexos.

Historia de usuario	
<b>Número:</b> 1	<b>Nombre:</b> Visualizar recursos de aprendizaje.
<b>Usuario:</b> Profesor.	
<b>Prioridad en negocio:</b> alta	<b>Riesgo en desarrollo</b> medio
<b>Puntos estimados:</b> 2 días	<b>Iteración asignada:</b> 1
<b>Descripción:</b> El profesor al dar clic en la opción visualizar se observa los recursos de aprendizaje.	
<b>Observación:</b> El usuario debe estar autenticado con el rol profesor.	

Tabla 6: HU Visualizar recursos de aprendizaje.

Historia de usuario	
<b>Número:</b> 2	<b>Nombre:</b> Visualizar estilos de aprendizaje.
<b>Usuario:</b> Profesor.	
<b>Prioridad en negocio:</b> alta	<b>Riesgo en desarrollo</b> medio
<b>Puntos estimados:</b> 2 días	<b>Iteración asignada:</b> 1
<b>Descripción:</b> El profesor al dar clic en la opción visualizar se le muestran los estilos de aprendizaje por estudiantes.	
<b>Observación:</b> El usuario debe estar autenticado con el rol profesor.	

Tabla 7: HU Visualizar estilos de aprendizaje

Historia de usuario	
<b>Número:</b> 3	<b>Nombre:</b> Asignar estilos a los recursos.
<b>Usuario:</b> Profesor	
<b>Prioridad en negocio:</b> alta	<b>Riesgo en desarrollo:</b> medio
<b>Puntos estimados:</b> 3 días	<b>Iteración asignada:</b> 2
<b>Descripción:</b> Cuando se da clic en la opción de “Asignar recursos” se permite poner a cada recurso el estilo de aprendizaje para el cual fue diseñado.	
<b>Observación:</b> El usuario debe estar autenticado con el rol profesor. Se puede asignar el mismo recurso para varios estilos de aprendizaje.	

Tabla 8: HU Asignar estilos a los recursos.

Para la realización del presente trabajo se tuvo en cuenta la duración de las semanas en las estimaciones de las historias de usuarios anteriores. Es importante mencionar que las semanas laborales están compuestas de 5 días ya que los sábados y los domingos no se trabajan. Un cálculo erróneo es contar la semana basándose en 7 días, esto llevaría a una mala planificación del trabajo. A continuación, se muestra una tabla donde se analizaron los días laborales de un mes escogido al azar como ejemplo.

Mes ejemplo						
<b>Domingo</b>	<b>Lunes</b>	<b>Martes</b>	<b>Miércoles</b>	<b>Jueves</b>	<b>Viernes</b>	<b>Sábado</b>
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

	Fin de semana		Próximo mes		Días laborales
--	---------------	--	-------------	--	----------------

Tabla 9: Mes ejemplo de días laborales

## 2.4 – Planificación

En la metodología XP, la creación del sistema se divide en iteraciones. La duración ideal de una iteración está entre una y tres semanas. Para cada una de las iteraciones el cliente establece un conjunto de HU que serán implementadas en cada iteración del sistema. Al final de cada iteración se realizan las pruebas de aceptación y la aplicación tendrá implementadas funcionalidades para dar cumplimiento a los objetivos propuestos (Letelier, 2008).

**2.4.1– Plan de iteraciones**

<i>Iteración</i>	<i>Historia de Usuario</i>	<i>Prioridad</i>
1	Visualizar recursos de aprendizaje. Visualizar estilos de aprendizaje. Visualizar la interfaz de usuario. Configurar módulo.	Alta
	Activar o desactivar módulo.	Baja
2	Asignar estilos a los recursos. Calcular experiencia de usuario.	Alta
	Generar reporte.	Medio
3	Calcular estilos de aprendizaje. Modificar la ruta del estudiante. Modificar visibilidad de los recursos. Definir ponderación, pesos de los recursos y actividades.	Alta

Tabla 10: Plan de iteraciones

Para realizar el plan de iteraciones se tuvo en cuenta el tiempo estimado que demora implementar cada HU además se necesita tener en cuenta la prioridad que el cliente le asignó a cada HU y la complejidad que posee la misma.

<b>Iteración</b>	<b>Historia de Usuario</b>	<b>Duración Total (en semanas)</b>
1	Visualizar recursos de aprendizaje. Visualizar estilos de aprendizaje. Visualizar la interfaz de usuario. Configurar módulo. Activar o desactivar módulo.	1 semana y 3 días
2	Asignar estilos a los recursos. Generar reporte. Calcular experiencia de usuario.	1 semana y 3 días
3	Calcular estilos de aprendizaje. Modificar la ruta del estudiante. Modificar visibilidad de los recursos. Definir ponderación, pesos de los recursos y actividades.	1 semana y 6 días

Tabla 11: Duración total (en semanas) de las iteraciones

### 2.4.2– Plan de entregas

El plan de entregas es el compromiso final del equipo de desarrollo con los clientes. 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 de software 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 (Letelier, 2008).

<b>Historia de usuario</b>	<b>1ra iteración 28 de marzo al 8 de abril</b>	<b>2da iteración 11 al 22 de Abril</b>	<b>3ra iteración 25 de abril al 11 Mayo</b>
Visualizar recursos de aprendizaje.	V 1.0	F	
Visualizar estilos de aprendizaje.	V 1.0	F	
Visualizar la interfaz de usuario.	V 1.0	F	
Configurar módulo.	V 1.0	F	
Activar o desactivar módulo.	V 1.0	F	
Asignar estilos a los recursos.	-	V 1.1	F
Generar reporte.	-	V 1.1	F
Calcular experiencia de usuario.	-	-	V 1.2
Calcular estilos de aprendizaje.	-	-	V 1.2
Modificar la ruta del estudiante.	-	-	V 1.2
Modificar visibilidad de los recursos.	-	-	V 1.2
Definir ponderación, pesos de los recursos y actividades.	-	-	V 1.2

Tabla 12: Plan de entrega.

## **2.5– Conclusiones**

- Se describió minuciosamente la solución propuesta para ello se elaboró un modelo conceptual obteniendo así un mejor entendimiento de los conceptos relacionados con la solución.
- A partir de las HU se definieron los tipos de actividades, así como la determinación de la experiencia de usuario y de los estilos de aprendizaje, los cuales son clave para la correcta elaboración de la solución propuesta.
- El estudio del personal a quien va dirigida la solución definió que existen 3 tipos de usuarios que interactúan con el módulo, aunque tienen diferentes niveles de acceso los cuales son: estudiante, profesor y administrador.
- Se definieron 3 iteraciones que abarcan un total de 12 HU, las cuales describen los aspectos principales a tener en cuenta para desarrollar la solución.
- Se generó el plan de duración de las iteraciones que enmarcó el tiempo de desarrollo de las HU en 4 semanas y 5 días, se determinó un cronograma que especifica las entregas con las fechas en las que será liberada cada versión del sistema.

## **Capítulo 3: Diseño, implementación y prueba**

### ***Introducción***

La metodología XP, no propone concisamente los artefactos a utilizar en la implementación, deja en manos del equipo de desarrollo la decisión de utilizar tantos tipos de diagrama UML como crean posible y así facilitar el proceso de desarrollo. En este capítulo se describe la fase de diseño, implementación y prueba según XP. Se identifican y organizan las clases relevantes para las funcionalidades del sistema, las tareas de ingeniería generadas por cada historia de usuario, las tarjetas Clase-Responsabilidad-Colaboración (CRC) así como la arquitectura y los patrones arquitectónicos empleados en el desarrollo de la aplicación. Además, se realizará la descripción de diseño de base de datos y se evaluará la calidad de la aplicación a través del proceso de pruebas utilizado.

### ***3.1 – Patrón arquitectónico y de diseño***

La arquitectura en el nivel más sencillo se considera, la forma general de la estructura física aunque siendo más profundos es más que eso; es la manera que los distintos componentes se integran para lograr un todo cohesivo (Pressman, 2010).

Los patrones arquitectónicos ofrecen soluciones a problemas de arquitectura que se encuentran comúnmente en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor (Pressman, 2010).

Un patrón de diseño provee un diseño para refinar los subsistemas o componentes de un software, o la relación que existe entre ellos. Estos describen estructuras comúnmente recurrentes de componentes de comunicación que solucionan un problema general de diseño en un contexto particular (Tejeda González & Cabrera Torres, 2014).



### **3.1.2 – Patrones arquitectónicos**

Los patrones arquitectónicos especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes (Venete, 2011).

El desarrollo modular de Moodle ha contribuido a que el mismo no tenga una arquitectura definida, en su desarrollo se mezclan diferentes modelos arquitectónicos por lo que en nuevos desarrollos no es necesario seguir un patrón predeterminado. El patrón arquitectónico escogido para la elaboración del módulo es el Modelo-Vista-Controlador (MVC) ya que permite separar la lógica de negocio de la interfaz del usuario y con esto facilita la evolución por separado de ambos aspectos. Otros componentes utilizan este patrón que permite acceder a la Interfaz de programación de aplicaciones (API por sus siglas en inglés) de Moodle y mantener separadas las capas de presentación de las capas del modelo del negocio.

El MVC fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. MVC divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto utiliza las siguientes abstracciones (Bahit, 2007):

- **Modelo (Model):** encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- **Vista (View):** muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador (Controller):** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsaciones del ratón, pulsaciones de teclas, entre otros. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores.

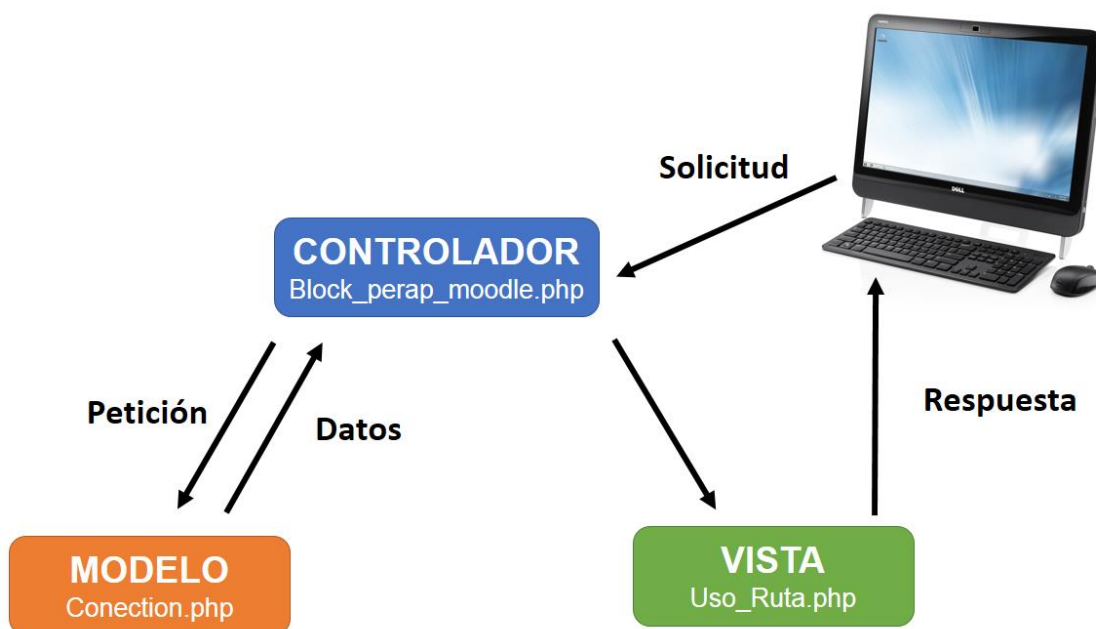


Fig 5: Modelo-Vista-Controlador

En la solución, cuando un profesor hace la petición que renderiza la vista `uso_ruta.php`, se hace un llamado a la función `$control->datos_graficos_uso_ruta()` en la clase controladora `Controler.php`. El controlador se encarga de solicitar al modelo, específicamente a la base de datos la información necesaria para llenar el reporte de uso de ruta de aprendizaje, una vez que la consulta se ha hecho y los resultados obtenidos se envían a la vista para que pueda ser construido el informe.

### 3.1.3 –Patrones de diseño

Los patrones de diseño no son más que una solución a problemas que ocurren de manera frecuente en el momento de empezar a programar o desarrollar. Según Nicolás Tedeschi<sup>4</sup> “Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” En otras palabras, brindan una solución ya probada y documentada a problemas de

---

<sup>4</sup> Profesor de la Universidad de la República del Uruguay, en la Facultad de Ingeniería.

desarrollo de software que están sujetos a contextos similares. Se deben tener presente los siguientes elementos de un patrón: su nombre, el problema (cuándo aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) (Tedeschi, 2014).

Dentro de los patrones del diseño más populares se encuentran los patrones de Asignación de Responsabilidades.

### **Patrones Asignación de Responsabilidades**

Los Patrones Generales de Asignación de Responsabilidades de Software (GRASP por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, de forma tal que se pueda diseñar software orientado a objetos. Los patrones GRASP están compuestos por: Experto, Creador, Bajo Acoplamiento, Alta Cohesión, Controlador, Fabricación Pura, entre otros (Jorge & Fuentes, 2015). Algunos de estos principios se tuvieron en cuenta durante todo el desarrollo del módulo propuesto.

**Experto:** se encarga de asignar la responsabilidad a la clase que cuenta con la información necesaria para ejecutar la tarea que tiene encomendada, lo cual se evidencia en la distribución de las clases como consta en el modelo-vista-controlador en el cual se apoya el diseño del módulo donde la clase controladora se encarga de manejar todo el flujo de datos e información concerniente al proceso y la vista de mostrar la misma todo esto procedente del modelo de datos definido.

**Creador:** tiene en cuenta para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto solo pueda ser creada por el objeto que contiene la información necesaria. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo cual favorece al mantenimiento del sistema. Este patrón se evidencia cuando se necesitan adquirir datos de la clase `controler.php` a medida que van haciendo falta ciertos datos.

**Alta cohesión:** propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de los límites manejables, para evadir un trabajo excesivo. Este patrón se puede

evidenciar en la clase controladora dado que está constantemente enlazada con métodos tanto propios como externos.

Fabricación pura: la fabricación pura se da en las clases que no representan un ente u objeto real del dominio del problema, sino que se ha creado con toda intención para disminuir el acoplamiento, aumentar la cohesión y potenciar la reutilización del código. Se crean clases de apoyo a modo de biblioteca personal que permite el trabajo con los datos tal como script.js.

### **3.2 – Diseño del sistema**

Para el diseño de las aplicaciones, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar utiliza técnicas como las tarjetas Clase, Responsabilidad y Colaboración (CRC). Además de las tarjetas CRC, para tener una idea clara y compacta de las clases y sus relaciones, se elaboró el diagrama de clases del diseño.

#### **3.2.1 – Tarjetas CRC**

El modelado Clase-Responsabilidad-Colaboración (CRC) proporciona una manera sencilla de identificación y organización de las clases que son relevantes para los requerimientos de un sistema o producto (Pressman, 2010).

Las tarjetas CRC son elaboradas durante la fase de diseño de la metodología de desarrollo XP, la cual constituye un modelo simple donde su objetivo principal es desarrollar una representación organizada de las clases. Cada tarjeta representa una clase, de la cual se describen las responsabilidades que tiene y las clases colaboradoras que se relacionan con la misma.

Una responsabilidad es cualquier cosa que la clase sabe o hace. Los colaboradores son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad (Beck & Fowler, 2000).

A continuación, se describen algunas tarjetas CRC, el resto se encuentran en los anexos:

Clase: Vista	
Responsabilidad	Colaboración
Se encarga de gestionar los datos relacionados a las clases de vistas.	Configuración Inicio Reporte Block_perap_moodle

Tabla13: Tarjeta CRC de Vista

Clase: Perap	
Responsabilidad	Colaboración
Se encarga los datos relacionados con la configuración del módulo.	Block_perap_moodle

Tabla14: Tarjeta CRC de Perap

Clase: Reporte	
Responsabilidad	Colaboración
Se encarga de generar los reportes relacionados a cada una de las opciones.	Uso_Recurso Uso_Ruta Uso_Estilos Trayectoria Exp_Usuario Block_perap_moodle

Tabla15: Tarjeta CRC de Reporte

### 3.2.2 – Diagrama de clases del diseño

Un diagrama de clases del diseño es un tipo de diagrama estático que describe la estructura de un sistema, mostrando sus clases, atributos y las relaciones entre ellos. Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará, los componentes que se encargarán del

funcionamiento y la relación entre uno y otro. Describen gráficamente las especificaciones de las clases de software y las interfaces (Pressman, 2010).

Aunque la metodología utilizada no exige la confección de este artefacto el equipo de desarrollo decidió elaborarlo para lograr un mejor entendimiento de la relación entre las clases.

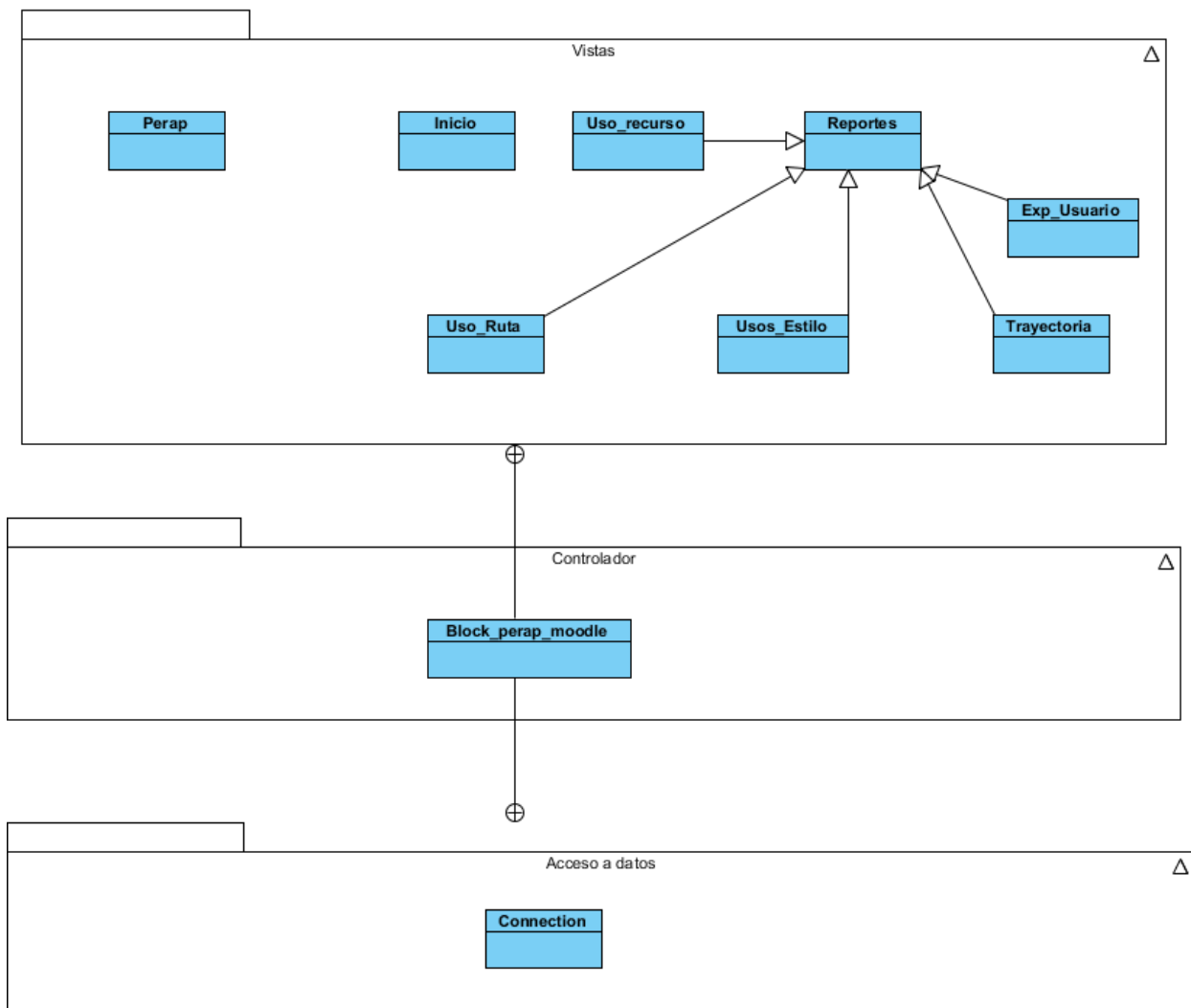


Fig. 6 – Diagrama de clases del diseño

### **3.3 – Diseño de la Base de Datos**

En cualquier sistema en el que se gestione información, la base de datos desempeña un papel fundamental. El diseño de la misma es un proceso esencial para el desarrollo de sistemas de este tipo, pues permite el almacenamiento de la información de forma coherente y organizada, evitando así pérdidas e inconsistencias, de esta manera la recuperación de los datos se puede realizar de forma rápida y flexible.

Para asegurar que los datos se mantengan de manera persistente y actualizada se introducen al modelo actual presentado por MOODLE 4 tablas de datos para manipular toda la información que genera y utiliza el módulo; las mismas serán descritas a continuación:

- mdl\_block\_perap\_moodle\_config: es donde se almacenan los valores de los campos requeridos para la configuración del módulo. Como el módulo no tiene por qué ser utilizado en todos los cursos esta tabla almacena para que cursos está activo. Los principales atributos de esta tabla son:
  - name (varchar): sirve para identificar la configuración a la que se hace referencia.
  - value (varchar): almacena el valor de la configuración asociada a la tupla
  - courseid (entero): representa el identificador del curso para el cual el módulo éste activo o haya estado en uso.
  - userid (entero): es el identificador del usuario al cual se le aplica la configuración requerida.
- mdl\_block\_perap\_moodle\_crs\_st: esta tabla se encarga de llevar el estado de los cursos para los cuales se encuentra activo o haya estado activo el módulo además de presentar la disponibilidad de los recursos del curso.
  - userid (entero): identifica el usuario para el cual se presentan los recursos
  - courseid (entero): representa el identificador del curso al que el usuario se ha inscrito
  - recourseid (entero): identificador único del recurso en la tabla mdl\_block\_perap\_moodle\_rss.
  - mostrar (varchar): denota cuál de los recursos se ha de mostrar y cuáles no.

- usado (varchar): denota qué recurso ha sido usado y cual no.
- mdl\_block\_perap\_moodle\_rss: es la tabla encargada de listar todos los recursos pertenecientes al curso para poder guardar los datos referentes a los mismos y al curso al que pertenecen.
  - resourceid (entero): identifica de manera única cada uno de los recursos asociados al curso.
  - name\_resource (varchar): indica el nombre del recurso.
  - courseid (entero): indica el identificador del curso para el cual está activo o haya estado el modulo.
  - name\_table (varchar): indica la tabla que utiliza la plataforma para gestionar la información de cada uno de los recursos.
  - style(1-4) (varchar): que asume la tarea de denotar si un recurso pertenece a un estilo determinado por medio de asignaciones binarias.
  - ponde (entero): guarda el nivel de importancia de los recursos.
- mdl\_block\_perap\_moodle\_usr\_st: es la tabla encargada de gestionar el estado referente a los usuarios que se encuentran en un curso con el módulo activo.
  - userid (entero): identifica los usuarios que se encuentran en un curso con el módulo activo o hayan sido parte de un curso con esta característica.
  - courseid (entero): representa el identificador del curso al que el usuario se ha inscrito
  - style(1-4) (varchar): guarda el valor que define la probabilidad de que un tipo de recurso sea el correcto para el estudiante.
  - laststyle (entero): define el último estilo propuesto para el estudiante.

Moodle tiene para su funcionamiento una base de datos con más de 300 tablas. A continuación se muestra el modelo físico de la base de datos que se utilizó teniendo en cuenta las tablas que genera el módulo y las tablas de Moodle con las que se tiene relación.



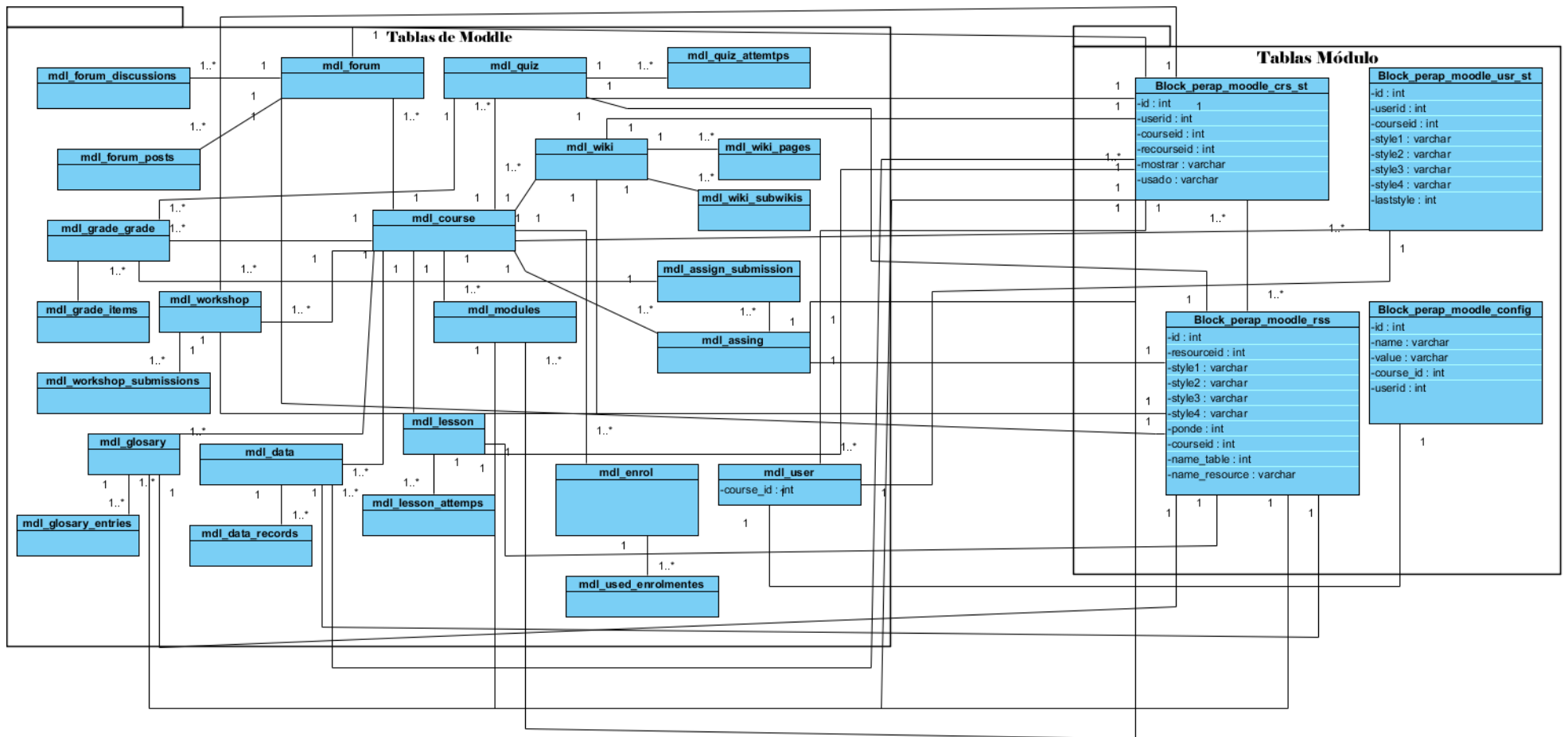


Fig. 7 – Diagrama de diseño de la Base de Datos

### 3.3 – Implementación del sistema

Para describir las tareas llevadas a cabo en la fase de implementación, se emplea un lenguaje técnico, el cual, no necesariamente debe ser entendible por el cliente. Dichas tareas son asignadas al equipo o programador responsable, normalmente la codificación se lleva a cabo por una pareja de programadores. Esta labor se lleva a cabo con el objetivo de detallar mejor las historias de usuario, lo cual facilita el entendimiento en el proceso de implementación. Cada historia de usuario puede contener una o más tareas de ingeniería, explicando de forma general las acciones que se realizan en la misma.

#### 3.4.1 – Tareas de ingeniería

La metodología XP propone dividir cada HU en tareas de ingeniería, cuyo objetivo fundamental es facilitar la implementación al programador (Canós, Leteriel, & Penadés, 2003).

Las tareas de ingeniería son escritas por el equipo de desarrollo a partir de las Historias de Usuario elaboradas por el cliente, brindando un detalle más profundo para realizar una implementación de las mismas y estimando un tiempo más cercano a la realidad para cada una de ellas (Beck & Fowler, 2000).

A continuación se detallan las tareas de la ingeniería más importantes para el módulo, el resto se encuentran en los anexos.

Tarea de ingeniería	
<b>Número:</b> 1	<b>Número de HU:</b> 1
<b>Nombre:</b> Visualizar recursos de aprendizaje.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 días
<b>Fecha inicio:</b> 28 de marzo	<b>Fecha fin:</b> 30 de marzo
<b>Programador (es) responsable:</b> Luciano Pumeda Ale, Edelin Cruz Febles	
<b>Descripción:</b> El profesor al dar clic en la opción visualizar le muestra una lista de los recursos subidos al curso, a la derecha del mismo se encuentran las opciones: seleccionar estilos de aprendizaje y seleccionar nivel de importancia.	

Tabla 16: Tarea de ingeniería número 1.

Tarea de ingeniería	
<b>Número:</b> 2	<b>Número de HU:</b> 2
<b>Nombre:</b> Visualizar estilos de aprendizaje.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 días
<b>Fecha inicio:</b> 31 de marzo	<b>Fecha fin:</b> 2 de abril
<b>Programador (es) responsable:</b> Luciano Pumeda Ale, Edelin Cruz Febles	
<b>Descripción:</b> El profesor al dar clic en la opción visualizar reporte se muestra un reporte que detalla que cantidad de estudiantes tributan a cada estilo de aprendizaje, la información se muestra en forma de gráfico.	

Tabla 17: Tarea de ingeniería número 2.

Tarea de ingeniería	
<b>Número:</b> 3	<b>Número de HU:</b> 4
<b>Nombre:</b> Visualizar la interfaz de usuario.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2días
<b>Fecha inicio:</b> 2 de abril	<b>Fecha fin:</b> 4 de abril
<b>Programador (es) responsable:</b> Luciano Pumeda Ale, Edelin Cruz Febles	
<b>Descripción:</b> El usuario profesor una vez autenticado en el curso selecciona el módulo y accede a la interfaz de usuario, siendo recibido por la pantalla de inicio de la misma, la cual cuenta con una breve descripción del módulo.	

Tabla 18: Tarea de ingeniería número 3.

Tarea de ingeniería	
<b>Número:</b> 4	<b>Número de HU:</b> 5
<b>Nombre:</b> Configurar módulo.	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 días
<b>Fecha inicio:</b> 4 de abril	<b>Fecha fin:</b> 7 de abril
<b>Programador (es) responsable:</b> Luciano Pumeda Ale, Edelin Cruz Febles	
<b>Descripción:</b> El profesor selecciona la opción de configurar el módulo y accede a la interfaz de configuración.	

Tabla 19: Tarea de ingeniería número 4.

### 3.5 – Diagrama de despliegue

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes: hardware y software, en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

A continuación se muestra el diagrama de despliegue realizado para la correcta elaboración del módulo.

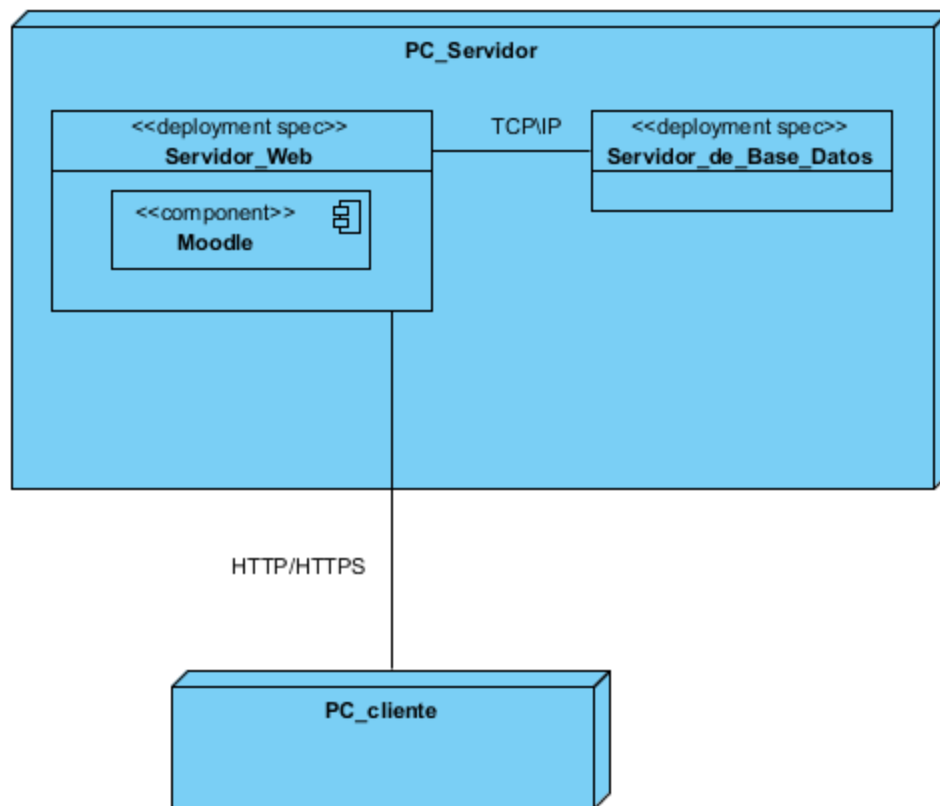


Fig 8: Diagrama de Despliegue

Nombre del nodo: Pc\_Cliente: Este nodo hace referencia a cualquier estación de trabajo (Pc\_Cliente) desde la cual se accede al sistema vía web.

Nombre del nodo: PC\_servidor se encarga de procesar todas las peticiones del usuario, por lo que los requisitos de hardware en este caso son mínimos, aunque en cuanto a software hay que tener en cuenta que se necesita instalar la plataforma educativa Moodle en el servidor.

### **3.6 – Diagrama de componentes**

El diseño del diagrama de componentes para el software describe por completo los detalles internos de cada componente. Para lograrlo, este diseño define estructura de datos para todos los objetos de datos locales y detalles algorítmicos para todo el procesamiento que tiene lugar dentro de un componente, así como la interfaz que permite el acceso a todas las operaciones de los componentes (comportamientos) (Pressman, 2010).

A continuación se muestra el diagrama de componentes del módulo y la descripción de los principales elementos que lo conforman.

- **Paquete Vista:** contiene los componentes pertenecientes a las clases vistas.
- **Paquete Controlador:** contiene los componentes pertenecientes a la clase controladora.
- **Paquete Modelo:** contiene los componentes pertenecientes a la clase encargada de realizar la conexión a la base de datos.
- **Componente CCS:** contiene los archivos que garantizan la correcta presentación del módulo.
- **Biblioteca JQuery:** para interactuar de forma dinámica con los componentes visuales del curso de Moodle y realizar consultas con Ajax.
- **Biblioteca Angular:** se utiliza para la construcción visual de los reportes.
- **Componente Base De Datos:** encapsula todos los datos del sistema.

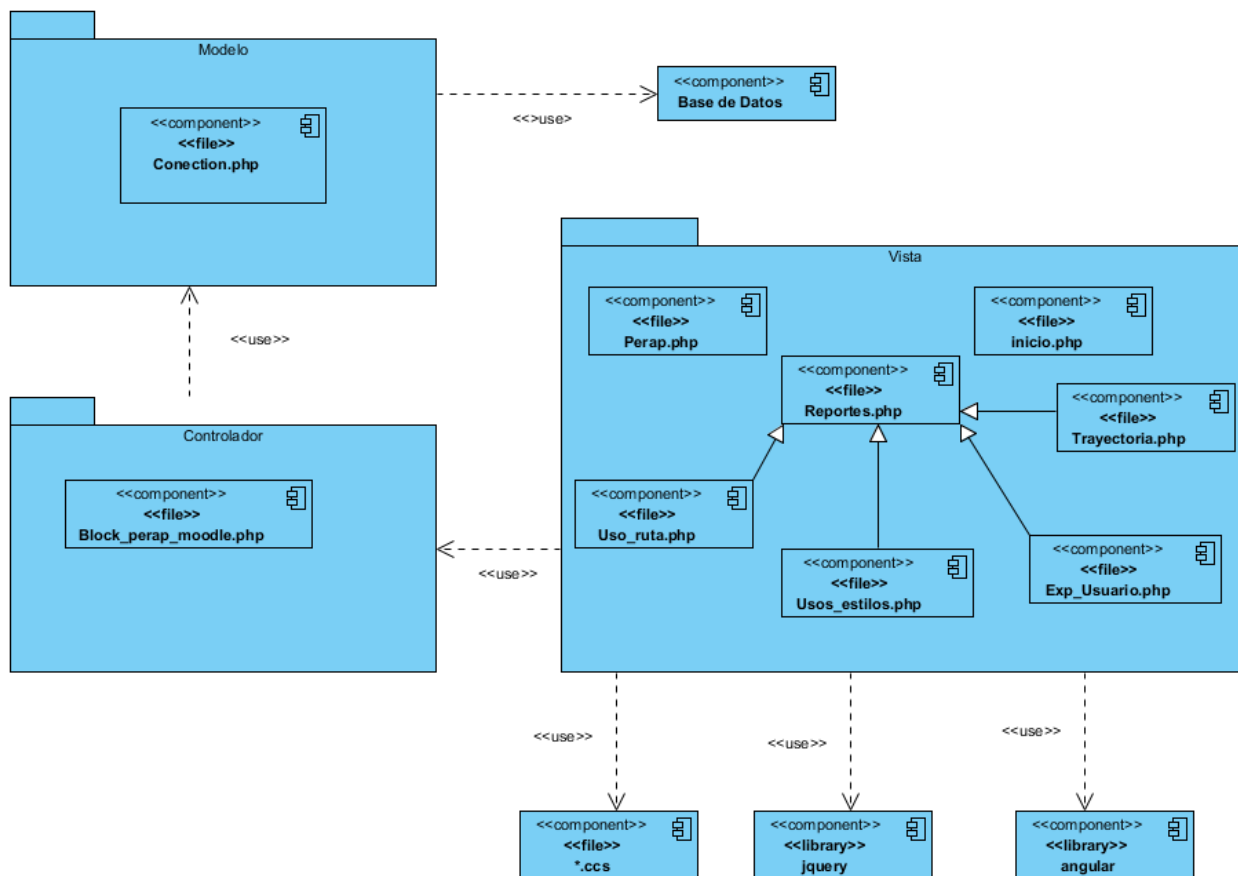


Fig 9: Diagrama de Componentes

### 3.7– Pruebas

Uno de los pilares de la metodología XP es el proceso de pruebas. XP anima a probar tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió las funcionalidades requeridas por el cliente final ( Gutiérrez, Escalona, Mejías, & Tor, 2010).

### 3.7.1– Pruebas Unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test-driven programming”). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas deben ser guardados junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo (Joskowicz, 2008).

Para las pruebas unitarias se usó el PHPUnit, el cual es uno de los principales frameworks de pruebas unitarias en PHP, que permite realizar las pruebas pertinentes al código, verificando que el funcionamiento de las aplicaciones PHP es el deseado encontrando bugs y errores que una vez solucionados mejorarán la calidad del desarrollo web (Eguiluz, 2012).

Moodle entre sus opciones de desarrollo trae incorporado la herramienta de pruebas unitarias PHPUnit, con la cual se realizaron dichas pruebas en el código del módulo.

A continuación se muestra un ejemplo del código donde se evidencia como fueron implementadas las pruebas unitarias:

```
public function test_active_nulo() {
    $active = $this->set_active();
    $this->update_active($active);
    $this->expect_exception_without_halting("block_perap_invalid_active_exception");
    unset ($active);
}
```

Fig 10: Implementación de la prueba unitaria del método test\_active\_nulo()

Al ejecutar las pruebas se muestra una consola con el resultado de la misma:

```
+{38;42mOK <21 tests, 21 assertions>+{0m
C:\wamp>php phunit.phar -c www\moodle\blocks\perap_moodle\tests\Entity
PHPUnit 4.1.0 by Sebastian Bergmann.
Configuration read from C:\wamp\www\moodle\blocks\tests\phpunit.xml.dist
.
Time: 310 ms, Memory: 9.25Mb
There was 1 failure:

1)perap_moodle\test\Entity\HintTest::testValidateActive
Error validando el hint, no debe ser nulo
Failed asserting that 1 matches expected 0.

C:\wamp>www\moodle\blocks\perap_moodle\tests\Entity\HintTest.php:37

+{37;41m                                     +{0m
+{37;41mFAILURES!                               +{0m
+{37;41mTests: 21, Assertions: 21, Failures: 1.+{0m
```

Fig 11: Resultado de la prueba unitaria del método test\_active\_nulo()

### 3.7.2– Pruebas de Aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada (Joskowicz, 2008).

Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información (Joskowicz, 2008).

A continuación se muestran dos casos de pruebas de aceptación, el resto se encuentran en los anexos:



Caso de Prueba	
<b>Número del Caso de Prueba:</b> 1	<b>Número de la HU:</b> 1
<b>Nombre de HU:</b> Visualizar recursos de aprendizaje.	
<b>Condiciones de Ejecución:</b> El usuario que tiene iniciada la sesión debe estar autenticado como profesor.	
<b>Pasos de ejecución/entrada:</b> Al dar clic en la opción visualizar le muestra una lista de los recursos subidos al curso, a la derecha del mismo se encuentran las opciones: seleccionar estilos de aprendizaje y seleccionar nivel de importancia.	
<b>Resultado esperado:</b> Vista todos los recursos de aprendizaje del curso.	

Tabla 20: Caso de Prueba

Caso de Prueba	
<b>Número del Caso de Prueba:</b> 2	<b>Número de la HU:</b> 2
<b>Nombre de HU:</b> Visualizar estilos de aprendizaje.	
<b>Condiciones de Ejecución:</b> El usuario que tiene la sesión iniciada debe estar autenticado como profesor.	
<b>Pasos de ejecución/entrada:</b> Al dar clic en la opción visualizar reportes se muestra un reporte que detalla que cantidad de estudiantes tributan a cada estilo de aprendizaje, la información se muestra en forma de gráfico.	
<b>Resultado esperado:</b> Vista con los cuatro estilos de aprendizaje definidos.	

Tabla 21: Caso de Prueba

### 3.7.3– Registro de no conformidades

Al realizar el proceso de pruebas es normal encontrar un grupo de no conformidades, las cuales se traducen en errores encontrados y funcionalidades no deseadas por el cliente. Al finalizar cada iteración se realizan pruebas y se muestra al cliente una versión funcional del software de forma que pueda detectar no conformidades en los requisitos implementados hasta el momento.

Como parte de la metodología XP, las no conformidades encontradas en cada iteración son las primeras tareas a resolver de la iteración siguiente, siendo el cliente el encargado de ordenarlas por prioridad. Algunas de ellas al no ser críticas, son arrastradas a la siguiente iteración dando un mayor grado de prioridad a las no conformidades más significativas para el cliente. Llevando a cabo este proceso se logran minimizar los niveles de aceptación de errores y un mayor grado de satisfacción por parte del cliente. De esta manera quedaron resueltas las no conformidades detectadas en la aplicación desarrollada.

El presente trabajo está dividido en tres iteraciones, a continuación se muestran las no conformidades encontradas en cada una de ellas:

Iteración	Descripción
Primera	<ol style="list-style-type: none"> <li>1. No se visualizan los recursos de aprendizaje de forma correcta en Configuración en la pestaña de Recurso ya que la tabla que contiene los mismos se desplaza en algunos de los navegadores hacia los lados.</li> <li>2. La configuración del módulo no es consistente, porque la opción de guardado no se activaba, como cuando se hacían cambios en las opciones y se volvían a desmarcar sin haber guardado los mismos.</li> <li>3. Faltas de ortografía evidenciadas en los acentos y en los mensajes mostrados como por ejemplo el nombre de la pestaña Configuración no mostraba la tilde.</li> <li>4. La validación de seguridad de acceso al módulo no está completa dado que cualquier usuario puede acceder al mismo.</li> <li>5. Texto en inglés en los mensajes del sistema en la pestaña Recurso perteneciente a la Configuración la cual dice "Quantity must be between 1-3" cuando debe decir "La cantidad supera el intervalo 1-3".</li> <li>6. La interfaz Inicio está fuera de rango, se pierde contenido hacia la derecha.</li> <li>7. La función de cálculo de experiencia de usuario bloquea el navegador al seleccionar la interfaz de Reportes en la pestaña de experiencia de usuario.</li> </ol>
Segunda	<ol style="list-style-type: none"> <li>1. El método mi_cron() no accede a la base de datos por lo cual no calcula las probabilidades.</li> <li>2. La fórmula de cálculo de experiencia de usuario en ocasiones no se acomoda a los datos obtenidos cuando se añaden recursos que no se hayan tenido en consideración, como es el caso de los módulos nuevos que se añadan en versiones nuevas de la plataforma.</li> <li>3. El reporte Uso de ruta no carga los datos dado que el formato de los mismos no es el que espera, siendo html y espera json.</li> <li>4. El símbolo % no aparece en las gráficas de pastel referente a los gráficos que especifican el uso de ruta y uso de estilos de aprendizaje.</li> <li>5. Más de 10 usuarios provoca que los que superan esta cantidad no aparezcan cuando se llenan los reportes.</li> <li>6. La palabra Cálculo en el reporte de experiencia de usuario no está acentuada.</li> </ol>
Tercera	<ol style="list-style-type: none"> <li>1. No se encuentran no conformidades en el módulo.</li> </ol>

Tabla 22: Registro de no conformidades.

### 3.8 – Validación de la investigación

Para la validación de la investigación se propuso obtener datos a partir de dos ediciones del mismo curso a distancia impartido en el CENED, uno sin utilizar el módulo y otro con el módulo en funcionamiento. Esta comparación permitió obtener datos estadísticos de uso de los recursos, interés de los usuarios, etc.

El curso escogido para realizar la validación fue “Los Recursos Educativos en la Educación a Distancia” en sus ediciones cuarta y quinta, el mismo cuenta con una gran variedad de recursos elaborados por los profesores que permitieron tener en cuenta los diferentes estilos de aprendizaje sin hacer grandes cambios en su diseño. Además en este curso se aplica al finalizar una encuesta de satisfacción a los estudiantes para determinar el grado de aceptación de los recursos, actividades, trabajo de los profesores y otros criterios. La encuesta aplicada se encuentra en los anexos. Para ambas ediciones se trabajó con matrícula de la Universidad, lo que permite eliminar la influencia de la conectividad en las tasas de abandono de los cursos.

A continuación se muestra una tabla con los datos de las dos ediciones del curso:

Edición	Cant. estudiantes	Aprobados	Suspensos	Abandonaron
4ta	21	6	2	13
5ta	19	9	2	8

Tabla 23: Datos de las ediciones del curso.

El curso está estructurado en tres temas y una tarea final. Se consideró abandono a todos los estudiantes matriculados que no completaron las actividades hasta el tercer tema.

Cómo se puede apreciar en la tabla el 61% de los estudiantes de la cuarta edición del curso abandonaron, mientras que en la quinta fue solo un 42.1%. En comparación con la cuarta edición los estudiantes manifestaron ver un curso más a la medida de sus necesidades.

El 99% de los estudiantes consideran que los materiales proporcionados han sido válidos y adecuados para el curso. El 100% planteó que la forma de organizar y presentar los contenidos ha facilitado su comprensión, que se ha aumentado su motivación y que a lo largo del desarrollo del curso no han tenido la intención de abandonarlo.

### **3.9 – Conclusiones**

- El diseño arquitectónico es la base para el óptimo desarrollo de un sistema, un deficiente diseño acarrea que el proceso de ejecutar actualizaciones y mejoras al mismo supone una pérdida considerable de tiempo y otros recursos. La presente investigación respalda su diseño sobre patrones bien establecidos, los cuales fueron escogidos en dependencia de su función, teniendo en cuenta a la hora de estructurar los componentes del producto.
- Las tareas de ingeniería sirvieron como soporte organizativo en el desarrollo de la codificación del software propuesto. Las mismas posibilitaron distribuir de forma organizada la programación entre los miembros del equipo de desarrollo.
- Al finalizar cada iteración fueron realizadas las pruebas de aceptación para fomentar la conformidad y seguridad del cliente con respecto al producto. Las pruebas aplicadas al sistema permitieron corregir a tiempo todas las no conformidades y garantizar un completamiento adecuado de cada uno de los componentes concebidos para el módulo.

## **Conclusiones**

Luego de la realización del módulo y del análisis de los resultados de la investigación se arriba a las siguientes conclusiones:

- El análisis del ambiente nacional e internacional, permitió reconocer la no existencia de una solución informática que diese soporte a las necesidades planteadas por el cliente.
- El módulo obtenido constituye una herramienta que permite la personalización del aprendizaje en los cursos que se imparten en la plataforma del CENED.
- La utilización de la metodología XP garantizó que el producto se desarrollara en el tiempo previsto, generando la documentación necesaria para ser utilizada como consulta en la realización de nuevas versiones del módulo.
- Las pruebas realizadas permitieron obtener un producto que cumple con las exigencias del cliente.

## **Recomendaciones**

Para el trabajo futuro los autores de esta tesis consideran que existen un grupo de aspectos que pudieran enriquecer la investigación, por lo que se propone:

- Los valores de ponderación utilizados en el trabajo responden a un conocimiento empírico de los especialistas del CENED por lo que se recomienda validar estos datos con el uso del propio módulo.
- Realizar un estudio sobre otros recursos y actividades de Moodle que pudieran tributar al cálculo de la experiencia de usuario.
- Las técnicas probabilísticas aplicadas a la teoría de juego constituye en este entorno una alternativa válida y de poco consumo de recursos para determinar los estilos de aprendizaje. Se recomienda comparar esta solución con las brindadas por los sistemas de hipermedia adaptativa.

## Referencias

- Basalo, A., & Alvarez, M. A. (28 de agosto de 2014). *desarrolloweb.com*. Recuperado el 6 de junio de 2016, de Qué es AngularJS: <http://www.desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html>
- Chapman, G. B. (24 de julio de 2005). *Short-term cost for long-term benefit: Time preference and cancer control*. Recuperado el 19 de noviembre de 2015, de PubMed: <http://www.ncbi.nlm.nih.gov/pubmed/16045418>
- Gutiérrez, J. J., Escalona, M. J., Mejías, M., & Tor, J. (2010). *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA*. Recuperado el 26 de abril de 2016, de [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/PSISEXTREMA.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf)
- Kahneman, D., & Tversky, A. (1992). *Advances in prospect theory: Cumulative representation of uncertainty*. *Journal of Risk and Uncertainty*.
- A Profile of the LMS Market. (17 octubre 2013.). *CampusComputing*, page 23.
- Aguiar, A. V. (2013). *Plugin Matriculación para la plataforma de teleformación Moodle en su versión 2.3.x.* . La Habana.
- Alfonso, A. S. (25 de junio de 2014). Inauguración del Centro Nacional de Educación a Distancia. Razones para dar un paso de transformación. *Periodico Gamma*.
- apache. (3 de marzo de 2016). *ABOUT\_APACHE*. Obtenido de [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)
- Bahit, E. (2007). *El paradigma de la Programación Orientada a Objetos en PHP con el patrón arquitectónico MVC*. Recuperado el 29 de abril de 2016, de <http://www.bubok.es/libros/205199/POO-y-MVC-en-PHP>
- Becerro, S. D. (Mayo de 2009). Plataformas educativas, un entorno para profesores y alumnos. Temas para la educación. *Revista digital para profesionales de la enseñanza*.
- Beck, K., & Fowler, M. (12 de octubre de 2000). *Planning Extreme Programming*. Obtenido de <https://ipfs.io/ipfs/QmTmMhRv2nh889JfYBWXdxSvNS6zWnh4QFo4Q2knV7Ei2B/Software%20Engineering/Addison-Wesley%20-%20Planning%20Extreme%20Programming.pdf>
- Brusilovsky, P. (2001). *Adaptive Hypermedia, User Modeling and User-Adapted Interaction*. Netherlands.: Kluwer Academic Publishers. Printed in the Netherlands.
- Canós, J. H., Leteriel, P., & Penadés, M. C. (2003). *Metodologías Ágiles en el desarrollo de software*. Valencia: Universidad Politécnica de Valencia.
- Castelló, J., Lerís, D., Martínez, V., & SeinEchalu. (2010). *Personalized Learning on the Moodle Platform using the CICEI Conditionals: Support Course in Mathematics INTED2010*

- Proceedings International Technology*. (Vols. ISBN:978-84-613-5538-9.). Valencia, España: IATED .
- Castillo, E. A. (2013). Los Sistemas Tutores Inteligentes y su impacto en la enseñanza de la programación. *Memorias del XV Congreso Internacional de Informática en la Educación, Informática*.
- Celia Rodríguez Ruiz, P. y. (19 de mayo de 2014). *edua y aprende*. Obtenido de <http://educayaprende.com/los-estilos-de-aprendizaje-las-diferentes-maneras-de-aprender/>
- Chris, S. (2000). Developments in non-expected utility theory: The hunt for a descriptive theory of choice under risk. *Journal of Economic Literature*,, (págs. 332-382).
- Clarenc, C. A., & S. M. Castro, C. L. (Diciembre de 2013). *Congreso Virtual Mundial de e-Learning*. Obtenido de Tosco Analizamos 19 plataformas de e- Learning: Investigación colaborativa sobre LMS. Grupo GEIPITE: [www.congresoellearning.org](http://www.congresoellearning.org)
- Curbelo, D. R. (2013). *Desarrollo del plugin External 2.0 para enlazar los objetos de aprendizaje del repositorio RHODA con Moodle v2.3.x*. La Habana.
- Díaz, E. (2012). Estilos de Aprendizaje. *Eídos*, 1-5. Obtenido de <https://www.google.com/cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUK EwiF8fCv7OjKAhVHej4KHTCaDb0QFggaMAA&url=http%3A%2F%2Ffiles.profesorraulgallego.webnode.com.co%2F200000314-6f19370135%2FESTILOS%2520DE%2520APRENDIZAJE.pdf&usq=AFQjCNG2zYm9r mXXtrSai>
- Equiluz, J. (2012). *Desarrollo Web Ágil con Symfony2*. Recuperado el 2 de mayo de 2016
- Freire, P. (1975). *La desmitificación de la conciencia y otros escritos*. Bogotá, Colombia: América Latina.
- Grimón, F., Guevara, M., & Ma Monguet, J. (2010). Influencia de usar un Sistema de Hipermedia Adaptativo (SHA) en la modalidad de Aprendizaje Combinado (Blended Learning). *Universidad Metropolitana.*, Vol. 10( N° 1 (Nueva Serie)), 93-112.
- Highsmith, J. (2002). *Agile Software Development Ecosystems*. Addison-Wesley.
- Ho, S. M., Mobini , S., Chiang , T.-J., Bradshaw , C., & Szabadi, E. (Octubre de 1999). *Publmed*. Recuperado el 11 de noviembre de 2015, de Theory and method in the quantitative analysis of "impulsive choice" behaviour: implications for psychopharmacology.: <http://www.ncbi.nlm.nih.gov/pubmed/10550487>
- Iglesias Sánchez , Á. (2013). *Modelo computacional cognitivo de toma de decisiones basado en el conocimiento: aplicación en la inferencia de explicaciones* . Madrid: Universidad Complutense de Madrid.



- Jacobson, I., Boch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*. Addison Wesley.
- Jorge, C. L., & Fuentes, N. R. (2015). *Sistema para la gestión de la Información de Ciencia Tecnología Innovación y Posgrado*. La Habana.
- Joskowicz, J. (2008). *Reglas y Prácticas en eXtreme Programming*. Vigo, España: Universidad de Vigo. Recuperado el 27 de abril de 2016, de <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- JQuery. (1 de abril de 2016). *JQuery*. Obtenido de <http://jquery.com/>
- Kahneman, D., & Tversky, A. (1984). *Choices, values, and frames*. *American Psychologist*.
- Kahneman, D., & Tversky, A. (1979). Prospect theory: an analysis of decision under risk. *Econometrica*, 263-292.
- Körding, K. (2007). *Decision theory: What should the nervous system do?* *Science*.
- Letelier, P. y. (2008). *Métodologías ágiles para el desarrollo de software:eXtreme Programming (XP)*. (Vols. Vol. 05, 26). Valencia: Universidad de Valencia.
- Lorenzatti, L. G. (23 de 2 de 2016). *blog net-learning*. Obtenido de "Personalizar el aprendizaje a través de las TIC": <http://www.net-learning.com.ar/blog/novedades-en-e-learning/personalizar-el-aprendizaje-a-traves-de-las-tic.html>
- Maguiña, M. I., & otros. (2015). *Rutas del aprendizaje*. Lima: Ministerio de Educación Perú.
- Moodle. (15 de diciembre de 2015). Obtenido de Community driven, globally supported.: <https://moodle.net/stats/>
- Moodle. (14 de diciembre de 2015). Obtenido de Community driven, globally supported.: [https://docs.moodle.org/all/es/Acerca\\_de\\_Moodle](https://docs.moodle.org/all/es/Acerca_de_Moodle)
- Moodle. (18 de enero de 2016). Obtenido de [https://docs.moodle.org/all/es/Arquitectura\\_de\\_Moodle](https://docs.moodle.org/all/es/Arquitectura_de_Moodle)
- NetBeans. (2016). *NetBeans*. Obtenido de [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html)
- Paisant, Y. L. (2011). *Extensión del módulo Chatxmpp de la plataforma de teleformación Moodle*. La Habana.
- Peña, C. I., Marzo, J.-L., Rosa, J. L., & ., R. F. (2012). Un sistema de tutoría inteligente adaptativo considerando estilos de aprendizaje. *Revista UIS Ingenierías. Universidad Industrial de Santander, Vol. 1* (Número 2. ).
- Pérez Mallea, I. (2015). *Los Entornos Personales de Aprendizaje*. La Habana.
- postgresql*. (21 de enero de 2016). Obtenido de [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)

- Prensky, M. (28 de 1 de 2016). *blog tiching*. Obtenido de "Debemos descubrir la pasión de cada estudiante": <http://blog.tiching.com/marc-prensky-debemos-descubrir-la-pasion-de-cada-estudiante/>
- Pressman, R. S. (2010). *Ingeniería de Software: Un enfoque práctico*. Distrito Federal (DF), México: Editorial Mexicana. Recuperado el 22 de 1 de 2016
- Rojas, Y. I., & Alfonso, O. F. (2015). *Sistema de Adaptación de Contenido basado en Estilos de Aprendizaje para Cursos Virtuales*. La Habana: UCI.
- Rumbaugh, J. J. ( 2000). *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison-Wesley.
- Rustichini, A., Stephen, V. B., Carpenter, J. P., & Götte, L. (2008). *Cognitive Skills Explain Economic Preferences, Strategic Behavior, and Job Attachment*. Germany : Forschungsinstitut zur Zukunft der Arbeit Institute for the Study of Labor.
- Sein-Echaluze, D. L. (3 de diciembre de 2011). "LA PERSONALIZACIÓN DEL APRENDIZAJE: UN OBJETIVO DEL PARADIGMA EDUCATIVO CENTRADO EN EL APRENDIZAJE". *ARBOR Ciencia, Pensamiento y Cultura*, Vol. 187( ISSN: 0210-19), 123-134.
- Sein-Echaluze, M. L., & Lerís, D. &. (2011). Diseño instruccional adaptativo de cursos online en Ingeniería. *Promotion and Innovation with New Technologies in Engineering Education (FINTDI)*. (págs. 1-8). Digital Object Identifier: 10.1109/FINTDI.2011.5945972.
- Tedeschi, N. (14 de julio de 2014). *microsoft.com*. Recuperado el 4 de 18 de 2016, de <https://msdn.microsoft.com/es-es/library/bb972240.aspx#mainSection>
- Tejeda González, Y., & Cabrera Torres, M. (2014). *Herramienta de autor para crear tutores inteligentes basados en la ejemplificació*. La Habana, Cuba: Univerdidad de las Ciencias Informáticas.
- tipos de educación formal no formal e informal*. (13 de 05 de 2009). Obtenido de [edurecblog.com](http://edurecblog.com)
- Universidad de Deusto, 2000 . (s.f.). Obtenido de [http://www.cca.org.mx/profesores/cursos/cep21/modulo\\_2/modelo\\_kolb.htm](http://www.cca.org.mx/profesores/cursos/cep21/modulo_2/modelo_kolb.htm)
- Vélez, J., & Fabregat, R. ( 2007). Arquitectura para la Integración de las. Dimensiones de Adaptación en un Sistema Hipermedia Adaptativo. *Published at Proceedings* . Obtenido de *Published at Proceedings*.
- Venete, A. (2011). *Introducción a los patrones de la arquitectura*. Ingeniería de Software. Universitat Jaume I (UJI). Recuperado el 28 de abril de 2016
- Versione. (2 de diciembre de 2015). *The 10 annual state of agile report*. Obtenido de <http://www.stateofagile.versione.com>
- Watson, W. R. (2013). "An Argument for Clarity: What are Learning Management Systems, What

are They Not, and What Should They Become?". *TechTrends*, 28–34.

Zapata-Ros, M. (2013). *eBook*. Obtenido de "Analítica de aprendizaje y personalización" e-LiS.e-prints in Library and information science.: <http://eprints.rclis.org/19490/>

