

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

Desarrollo de un Sistema Web para la enseñanza de Casos de Uso empleando la Técnica de Aprendizaje Cooperativo de Rompecabezas.

Tesis para optar el Título de Ingeniero Informático, que presenta el bachiller:

Gerardo Yatsen Chung Pinzás

ASESOR: Dr. Jose Antonio Pow Sang Portillo

Lima, Setiembre de 2013

FACULTAD DE
**CIENCIAS E
 INGENIERÍA**
 ESPECIALIDAD DE
 INGENIERÍA INFORMÁTICA

 PONTIFICIA
**UNIVERSIDAD
 CATÓLICA**
 DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO

TÍTULO: Desarrollo de un Sistema Web para la enseñanza de Casos de Uso empleando la Técnica de Aprendizaje Cooperativo del Rompecabezas.

ÁREA: Desarrollo de programas

PROPONENTE: Dr. Jose Antonio Pow Sang

ASESOR: Dr. Jose Antonio Pow Sang

ALUMNO: Gerardo Chung Pinzás

CÓDIGO: 199700343

TEMA N°: 477

FECHA: San Miguel, 18 de julio de 2013


DESCRIPCIÓN

La técnica del Rompecabezas (en inglés: *Jigsaw Teaching Technique*) es una técnica de aprendizaje cooperativo que se basa en la contribución de los estudiantes para alcanzar un objetivo común. Esta técnica propone la división de tareas o temas parciales de una materia. De esta forma, cada miembro del grupo tendrá que ejecutar una de estas tareas parciales, las cuales finalmente deberán ser integradas por todos los miembros en conjunto.

La técnica del Rompecabezas, fue desarrollada originalmente para la enseñanza en un salón de clases sin la participación de sistemas automatizados. Esta solución, a pesar de las ventajas que brinda, puede presentar algunas desventajas debido a la logística que requiere para implementarse.

Es por esos motivos que se propone el desarrollo de un Sistema Web que implemente la Técnica de Aprendizaje Cooperativo de Rompecabezas; el cual permitirá simplificar la implementación que se realizaban en un salón de clase, debido a la automatización y virtualización de sus procesos.

OBJETIVO GENERAL

El objetivo del presente proyecto es el desarrollar un sistema Web que permita simplificar los procesos de implementación de la técnica del Rompecabezas; permitiendo, de esta manera, su aplicación en la enseñanza del desarrollo de software, enfocándose específicamente en la elaboración cooperativa de casos de uso.

 Av. Universitaria 1801
 San Miguel, Lima - Perú

 Apartado Postal 1761
 Lima 100 - Perú

 Teléfono:
 (511) 626 2000 Anexo 4801

FACULTAD DE
**CIENCIAS E
 INGENIERÍA**
 ESPECIALIDAD DE
 INGENIERÍA INFORMÁTICA

 PONTIFICIA
**UNIVERSIDAD
 CATÓLICA**
 DEL PERÚ

OBJETIVOS ESPECÍFICOS

Los objetivos específicos del presente proyecto son:

- Realizar el análisis y diseño del sistema que cubra los requerimientos establecidos por la técnica del Rompecabezas.
- Elaborar una arquitectura del sistema que permita al sistema funcionar en navegadores Firefox, Chrome e Internet Explorer.
- Elaborar una herramienta que permita gestionar la participación cooperativa de los alumnos durante una sesión con la técnica del Rompecabezas.
- Definir y ejecutar las pruebas que permitan validar los requerimientos identificados.

ALCANCE

El sistema dará soporte a las fases que comprende una sesión de clase que emplea esta técnica, las cuales se dividen en los módulos de: Planificación, Ejecución y Evaluación.

Para el módulo de Planificación, el sistema permitirá al docente el registro de una dinámica asociada a un curso. El alumno podrá revisar la información asociada a la dinámica y tener acceso a la documentación seleccionada por el docente.

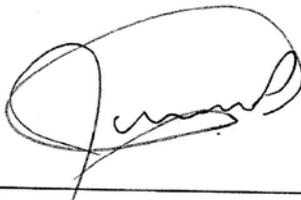
Para el módulo de Ejecución, el sistema proporcionará al alumno el acceso a una sesión cooperativa junto a los integrantes de su grupo inicial y del grupo experto al que esté asignado. También proporcionará al alumno el acceso a foros de discusión asignado a cada dinámica creada.

Para el módulo de Evaluación, el sistema permitirá al docente la creación de una evaluación asociada a la dinámica creada mediante el registro de enunciados, preguntas y respuestas; permitiendo a su vez la corrección manual y automática de las respuestas brindadas por los alumnos. El alumno tendrá acceso a un portal donde podrá rendir la evaluación, registrar sus respuestas y revisar las calificaciones obtenidas de las evaluaciones que haya formado parte.

El Sistema presentará un servicio de mensajería interna para la comunicación entre los Usuarios registrados.



Máximo: 100 páginas



 Av. Universitaria 1801
 San Miguel, Lima - Perú

 Apartado Postal 1761
 Lima 100 - Perú

 Teléfono:
 (511) 626 2000 Anexo 4801



DEDICATORIA

A mis padres y hermanos por su apoyo incondicional.

A mi asesor Jose Antonio Pow Sang por su paciencia y preocupación demostrada durante todo este proceso.

Y especialmente a Veronica Velez y Karina Gonzalez, con quienes inicié el desarrollo de la presente Tesis, y cuyos valiosos aportes forman parte importante de la base de este trabajo.

Karina, Veronica: LO LOGRAMOS !!

RESUMEN

Jigsaw es una técnica de aprendizaje cooperativo que se basa en la contribución de los estudiantes para alcanzar un objetivo común. Esta técnica también conocida como Técnica de Rompecabezas propone la división de tareas o temas parciales de una materia. De esta forma, cada miembro del grupo tendrá que ejecutar una de estas tareas parciales, las cuales finalmente deberán ser integradas por todos los miembros en conjunto.

La Técnica *Jigsaw*, fue desarrollada originalmente para la enseñanza dentro de un salón de clases sin la participación de sistemas automatizados. Esta solución, a pesar de las ventajas que brinda, puede presentar algunas desventajas debido a la logística que requiere para implementarse.

Es por esos motivos que se propuso el desarrollo de un Sistema Web que implemente la Técnica *Jigsaw*, el cual permitirá aprovechar conjuntamente las ventajas que brinda la comunicación vía Internet, así como las contribuciones propias de la técnica de aprendizaje cooperativo en mención.

El Sistema Web *Jigsaw* dará soporte a las tres fases que comprende una sesión de clase que emplea esta técnica, las cuales se dividen en los módulos de: Planificación, Ejecución y Evaluación. Estos módulos se encuentran estrictamente relacionados ya que siguen los procesos de diseño, desarrollo y evaluación de una Sesión de Clase *Jigsaw* respectivamente.

El módulo de Planificación tiene la principal responsabilidad de permitir el diseño de cada sesión de clase. Aquí se plantean los datos de la sesión que constituirán posteriormente la base de los objetivos y resultados esperados que permitirán medir el progreso académico de los alumnos en una etapa posterior.

El módulo de Ejecución se encarga de llevar a cabo la ejecución de una sesión de clase basada en la técnica del Rompecabezas. Permite el desarrollo paso a paso de la técnica desde la lectura de materiales, documentos y casos hasta la diagramación de la solución en los diferentes Grupos *Jigsaw* y Expertos. Además cuenta con herramientas de comunicación en línea, los cuales permitirán la interacción para la coordinación y elaboración de Casos de Uso.

Además se contará con Foros de Discusión para el desarrollo y esclarecimiento de algunas dudas sobre estos temas propuestos.

Finalmente el módulo de Evaluación permite la elaboración de preguntas y evaluaciones por parte del tutor y docente. Así mismo brinda la posibilidad de calificación manual y automática de las evaluaciones rendidas.

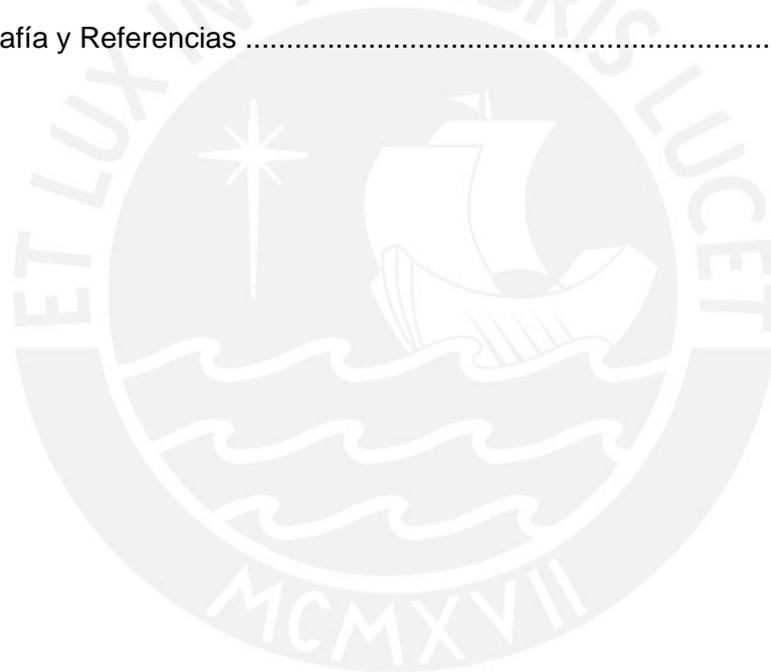
La presente tesis se desarrolló utilizando UML como lenguaje de modelamiento, RUP para los procesos a realizar durante el trabajo y el lenguaje de programación Java para el desarrollo del Sistema Web.



TABLA DE CONTENIDOS

Introducción	1
Capítulo I: Generalidades	4
1.1. Definición del problema	4
1.1.1. Técnica de Aprendizaje Cooperativo de Rompecabezas.....	4
1.1.2. Planteamiento del problema	6
1.1.3. Objetivo General.....	9
1.1.4. Objetivos Específicos	9
1.1.5. Resultados Esperados	11
1.2. Marco conceptual del problema	12
1.2.1. Aprendizaje Cooperativo	13
1.2.2. Técnica de Aprendizaje Cooperativo: <i>Jigsaw</i>	14
1.3. Plan del Proyecto	15
1.3.1. Métodos y procedimientos.....	15
1.4. Estado del Arte	18
1.4.1. Aplicaciones existentes	18
1.4.2. Evaluación de requerimientos Sistema Web <i>Jigsaw</i>	24
1.4.3. Requisitos del Sistema Web <i>Jigsaw</i>	26
Capítulo II: Análisis	28
2.1. Identificación de requerimientos	28
2.2.1. Requerimientos funcionales	28
2.2.2. Requerimientos no funcionales	35
2.2. Análisis de la solución	37
2.3.1. Evaluación de viabilidad y consideraciones sobre el sistema	37
2.3.2. Análisis Técnico.....	39
2.3.3. Definición del sistema.....	41
2.3.4. Requisitos Específicos: Especificación de Casos de Uso.....	48
2.3.5. Requisitos Específicos: Diagrama de Clases de Análisis	59
Capítulo III: Diseño	63
3.1. Arquitectura de la Solución.....	63
3.1.1. Metas Arquitectónicas y Restricciones.....	63
3.1.2. Arquitectura	65
3.2. Diseño de Interfaz Gráfica	70
3.2.1. Pantalla Principal.....	71
3.2.2. Pantalla del Módulo de Planificación	73
3.2.3. Pantalla del Módulo Ejecución	74
3.2.4. Pantalla del Módulo Evaluación	75

Capítulo IV: Construcción	77
4.1. Construcción.....	77
4.1.1. Patrones de Diseño	77
4.1.2. Patrones Core J2EE	79
4.1.3. Tecnología.....	80
4.1.4. Estándares de Programación	90
4.2. Pruebas	93
4.2.1. Estrategia de Pruebas	93
4.2.2. Pruebas de Aceptación	93
4.2.3. Pruebas de Seguridad.....	96
Capítulo V: Conclusiones y Trabajos Futuros.....	97
5.1. Conclusiones	97
5.2. Trabajos futuros.....	98
Bibliografía y Referencias	100



INDICE DE FIGURAS

Figura 1: Árbol de Problemas de la presente Tesis.	7
Figura 2: Árbol de Objetivos de la presente Tesis.....	10
Figura 3: Interfaz de Editor Cooperativo.....	20
Figura 4: Creación de Evaluaciones y Encuestas Campus Virtual PUCP I	22
Figura 5: Creación de Evaluaciones y Encuestas Campus Virtual PUCP II	22
Figura 6: Ejemplo de Evaluación en el Campus Virtual de la PUCP.....	23
Figura 7: Paquetes del Sistema	42
Figura 8: Diagrama de Casos de Usos del Paquete Planificación	43
Figura 9: Diagrama de Casos de Usos del Módulo de Ejecución	45
Figura 10: Diagrama de Casos de Usos del Módulo de Evaluación	47
Figura 11: Diagrama de Clases del Sistema Web Jigsaw.....	60
Figura 12: Vista de Casos de Uso.....	66
Figura 13: Vista Lógica.....	67
Figura 14: Diagrama de Componentes	69
Figura 15: Diagrama de Despliegue.....	70
Figura 16: Pantalla Principal	71
Figura 17: Pantalla Módulo de Planificación	73
Figura 18: Pantalla Módulo de Ejecución	74
Figura 19: Pantalla Módulo de Evaluación	76
Figura 20: Arquitectura de una Aplicación Struts 2	81
Figura 21: Arquitectura Servidor JBoss	86
Figura 22: Arquitectura MySQL	88

INDICE DE CUADROS

Cuadro 1: Descripción de Etapas e Iteraciones del Sistema Web Jigsaw	17
Cuadro 2: Evaluación de Funcionalidades	25
Cuadro 3: Requisitos Funcionales Modulo Planificación.....	30
Cuadro 4: Requisitos Funcionales Modulo Ejecución	32
Cuadro 5: Requisitos Funcionales Modulo Evaluación	34
Cuadro 6: Requisitos No Funcionales	36
Cuadro 7: Catálogo de Actores	41
Cuadro 8: Casos de Uso Paquete Planificación.....	44
Cuadro 9: Casos de Uso Paquete Ejecución	46
Cuadro 10: Casos de Uso Paquete Evaluación	48
Cuadro 11: Prefijos para notación de ubicación de variables	92
Cuadro 12: Prefijos para notación de tipo de dato de variables.....	92
Cuadro 13: Prefijos para notación de controles visuales	92



Introducción

La Técnica de Aprendizaje Cooperativo de Rompecabezas (*en inglés: Jigsaw Teaching Technique*) es una técnica de Aprendizaje cooperativo que se basa en la contribución de los estudiantes para alcanzar un objetivo común. Postulado por Aronson, esta propone la división de tareas o temas parciales de una materia. Estos temas son distribuidos a cada uno de los integrantes que conforman el Grupo *Jigsaw*. De esta forma, cada miembro del grupo tendrá que ejecutar una de estas tareas parciales, las cuales finalmente deberán ser integradas por todos los miembros en conjunto [1].

La práctica de esta técnica permite al estudiante desarrollar habilidades de comunicación ya que se ve en la necesidad de compartir, discutir y argumentar lo aprendido; lo cual contribuye a consolidar un aprendizaje más significativo y duradero [6]. Esta técnica ha sido implementada en los ambientes de clase del curso de Sistemas de Información 2 en la Pontificia Universidad Católica Del Perú como refuerzo a las clases magistrales impartidas previamente a cada tema del curso; la cual ha logrado resultados alentadores relacionados al nivel de aprendizaje de los estudiantes y un grado alto de empatía de los alumnos con la técnica [1].

A pesar de que esta técnica brinda grandes oportunidades a la calidad de instrucción universitaria, las tareas de planificación, ejecución y evaluación de una sesión de clase *Jigsaw* se tornan muy laboriosas [10][17][18]. Adicionalmente, se pueden presentar inconvenientes al efectuar una medición en la participación de los alumnos durante el desarrollo de dicha sesión [10].

Es por ello que la presente tesis presentará una propuesta de solución informática que dará soporte a las fases que comprende una sesión de clase que emplea la Técnica de Aprendizaje Cooperativo de Rompecabezas a través de un Sistema Web, el cual se conocerá con el nombre de Sistema Web *Jigsaw*, y estará distribuido en tres módulos: Planificación, Ejecución y Evaluación.

Si bien la Técnica de Aprendizaje Cooperativo de Rompecabezas es flexible y puede ser orientada a diversos temas a diferentes niveles de instrucción, este Sistema ha sido planteado en base a los temas de análisis y diseño orientado a

objetos; los cuales son temas esenciales del curso de Sistemas de Información 2, en los que anteriormente ya se había aplicado esta técnica.

En los capítulos presentados se describe el trabajo realizado para implementar la herramienta.

En el primer capítulo se presenta el marco teórico necesario con el fin de comprender los conceptos con los cuales se están trabajando. Estos conceptos están relacionados a la Técnica de Aprendizaje Cooperativo de Rompecabezas junto a otras técnicas de aprendizaje cooperativo utilizadas en la enseñanza actual, así como sus aplicaciones en diversos temas como el desarrollo de software. También se describe el escenario que abarca el problema el cual motivó el desarrollo del presente proyecto. Este escenario comprende los problemas encontrados en la implementación de dichas técnicas y la necesidad de emplear una herramienta automatizada que apoyará en la implementación de las distintas fases de la misma. Por último se presenta la solución propuesta por el Sistema Web *Jigsaw* el cual cubre los problemas planteados anteriormente.

El segundo capítulo corresponde a la etapa de análisis del Sistema propuesto. Se presenta un análisis de acuerdo a las necesidades del cliente y las restricciones del proyecto tanto a nivel económico como tecnológico. Así como un análisis a alto nivel del Sistema Web *Jigsaw* mediante la presentación del Catálogo de Requerimientos, los Casos de Uso más importantes del sistema y el Diagrama de Clases de Análisis.

En el tercer capítulo se describirá la etapa de diseño del Sistema Web *Jigsaw* la cual se enfoca directamente en la arquitectura de la solución y el diseño de interfaz gráfica. La primera sección en este capítulo está dirigida a la arquitectura de la solución la cual muestra las diferentes perspectivas, metas y restricciones a nivel técnico a los que estará expuesto el proyecto. A continuación, se describe las vistas de la arquitectura a utilizar en el desarrollo del sistema como base en las disciplinas que define RUP. Como segunda y última sección de este capítulo se hará un recorrido a través de las principales pantallas del sistema y se describirá su diseño y distribución.

El cuarto capítulo corresponde a las etapas de construcción y pruebas de la herramienta. Aquí se presentarán las especificaciones del prototipo y el

desarrollo de las principales funcionalidades. Luego se detallarán las especificaciones de diseño que se consideraron en el desarrollo del Sistema Web *Jigsaw*; así como, la presentación del Plan de Pruebas empleado, con el cual se garantiza el correcto funcionamiento del sistema.

Finalmente, en el quinto capítulo se expondrán los puntos que se concluyen de las fases de levantamiento de información, análisis, diseño e implementación del Sistema. Dentro del acápite de observaciones se explican detalles que se deben de tomar en cuenta sobre el desarrollo del proyecto. Así mismo, se brindarán las recomendaciones a tomar en cuenta para futuros trabajos que se desarrollen en base a la elaboración de un Sistema Web *Jigsaw*.



Capítulo I: Generalidades

Este capítulo presenta la descripción de la Técnica de Aprendizaje Cooperativo de Rompecabezas junto a otras técnicas de aprendizaje cooperativo utilizadas en la enseñanza actual, así como sus aplicaciones en diversos temas como el desarrollo de software. Así mismo describe los problemas encontrados en la implementación de dichas técnicas y la necesidad de emplear una herramienta automatizada que apoyará en la implementación de las distintas fases de la misma. Y por último se presenta la solución propuesta por el Sistema Web *Jigsaw* que cubrirá los problemas planteados anteriormente.

1.1. Definición del problema

La presente tesis buscó brindar una alternativa de optimización al empleo de la Técnica de Aprendizaje Cooperativo de Rompecabezas en la enseñanza del Desarrollo de Software, enfocándose específicamente en la elaboración cooperativa de Casos de Uso. En la sección 1.1.1 se brinda una breve introducción a la Técnica mencionada y a continuación, en la sección 1.1.2, se describen los problemas identificados en la enseñanza del Desarrollo de Software.

1.1.1. Técnica de Aprendizaje Cooperativo de Rompecabezas

Los métodos para el aprendizaje cooperativo brindan grandes ventajas a los alumnos ya que permiten aprovechar las habilidades de cada uno de los participantes para lograr una transmisión de conocimientos homogénea [3]. Este tipo de técnicas se basan en la cooperación entre estudiantes para alcanzar un objetivo común [6].

La esencia de los trabajos cooperativos es la participación en conjunto de los estudiantes. El empleo de este tipo de técnicas permite promover en los miembros del grupo un entendimiento más completo, significativo y duradero del tema a aprender [4]. Adicionalmente, los estudiantes mejoran sus habilidades de comunicación, ya que tienen que transmitir el conocimiento adquirido al resto de integrantes de equipo [5].

El método de Rompecabezas (*en inglés: Jigsaw*) es una de estas técnicas de aprendizaje cooperativo [10]. Esta técnica, que fue desarrollada inicialmente por

E. Aronson en la universidad de Texas y en la universidad de California en 1970, ha tenido resultados exitosos cuando ha sido utilizada.

La última versión de esta técnica fue desarrollada por Robert Slavin y es denominada como *Jigsaw 2* [7]. Esta versión consta de seis fases:

En la primera fase se deben desarrollar las siguientes actividades:

- Creación de Grupos: Estos deben estar compuestos por 3 a 4 estudiantes, de diferentes géneros y cualidades.
- Asignación del líder: De preferencia el líder debería tener experiencia en trabajos de grupo.
- Distribución de la lección: El número de partes debe estar de acuerdo con el número de integrantes del grupo.
- Distribución de materiales: Cada estudiante debe tener acceso a una sola sección de la lección
- Lectura Individual: Asignar un periodo para la lectura e investigación del material asignado

La segunda fase consta de la preparación de los grupos de “expertos”. Para lo cual cada uno de estos grupos debe tener asignado un tema a estudiar. Aquí los estudiantes deben leer y resumir su material de tal forma que al finalizar, deben ser “expertos” en dichos temas.

En la tercera fase cada miembro del grupo de expertos escucha la participación de cada uno de sus compañeros y realizan los comentarios, preguntas y/o sugerencias pertinentes relacionados al tema asignado. Así mismo practican la forma en la que enseñarán al resto de compañeros.

La cuarta fase de esta técnica consiste en poner en práctica todo lo aprendido. Los alumnos retornan a sus grupos originales (*Grupos Jigsaw*) y enseñan los temas que les fueron asignados. El objetivo de esta fase es lograr que todos aprendan todas las secciones que conforman el tema de la clase.

Durante el desarrollo de esta dinámica el profesor rotará de grupo en grupo, observará y asesorará a cada uno de los estudiantes (quinta fase).

Finalmente al término de la dinámica, se evaluará a todos los estudiantes individualmente la lección aprendida en clase (sexta fase).

1.1.2. Planteamiento del problema

El empleo de la Técnica de Aprendizaje Cooperativo de Rompecabezas no se restringe a ningún nivel de instrucción. Es por ello que puede ser empleada a nivel universitario. La Técnica de Aprendizaje Cooperativo de Rompecabezas ha sido implementada en los ambientes de clase del curso Sistemas de Información 2 en la Pontificia Universidad Católica Del Perú y ha logrado resultados alentadores relacionados al nivel de aprendizaje de los estudiantes [1] y un grado alto de empatía.

A pesar de que la Técnica de Aprendizaje Cooperativo de Rompecabezas brinda grandes oportunidades a la calidad de instrucción universitaria, el diseñar una sesión de clase empleando esta técnica puede ser una tarea bastante laboriosa para el docente y que puede generar problemas durante su implementación en un salón de clases. Por ejemplo, para el desarrollo de una sesión se genera un incremento de las actividades logísticas para la generación de material físico y para lograr tener la disponibilidad de un lugar físico apropiado para la ejecución de esta técnica; además se presenta la necesidad de gestionar el empleo de personal auxiliar que permita apoyar con la realización de las tareas de esta técnica; y finalmente el Docente muchas veces no se da abasto para atender o monitorear las consultas o necesidades de los alumnos durante la sesión.

Para realizar una evaluación más detallada sobre los problemas que se generan durante la implementación de esta técnica; se decidió emplear la Técnica de Árbol de Problemas.

La Técnica de Árbol de Problemas [8] es una técnica participativa que ayuda a desarrollar ideas creativas para identificar el problema y organizar la información recolectada, generando un modelo de relaciones causales que lo explican. Esta técnica facilita la identificación y organización de las causas y consecuencias de un problema. El tronco del árbol es el problema central, las raíces son las causas y la copa los efectos. A continuación en la Figura 1, se presenta el Árbol de Problemas referente a la implementación de la Técnica de Aprendizaje Cooperativo de Rompecabezas dentro de un Salón de Clase:

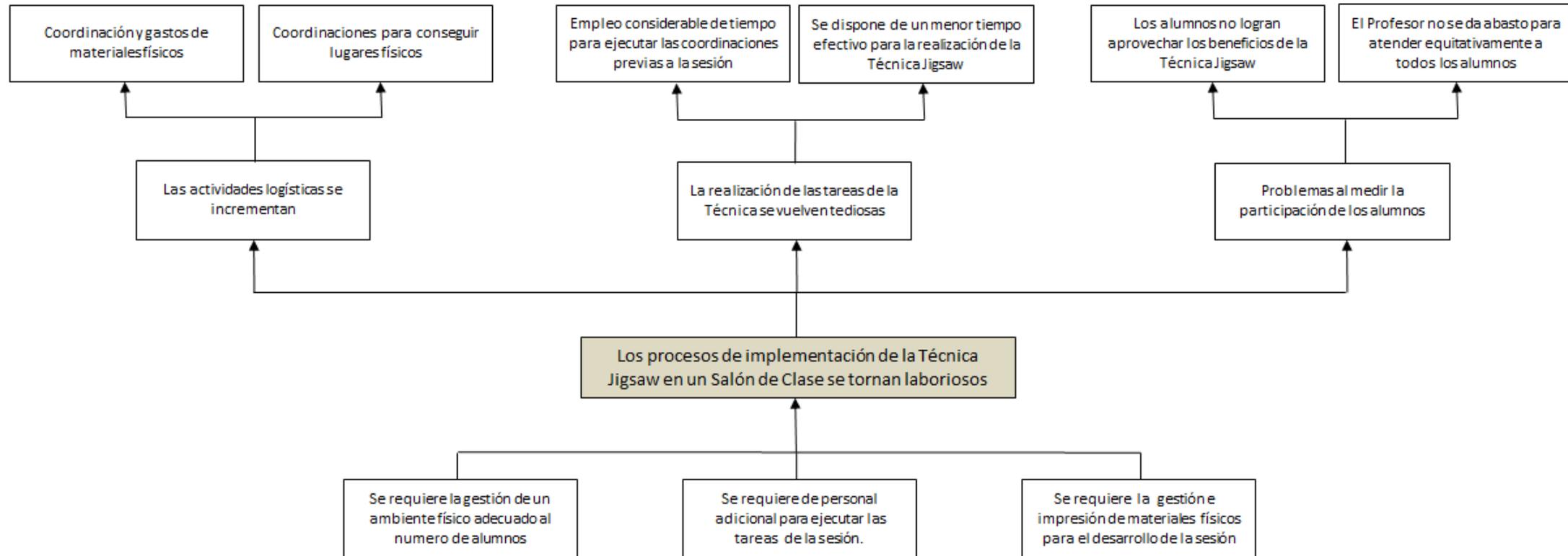


Figura 1: Árbol de Problemas de la presente Tesis.

Del Árbol de Problemas mostrado en la Figura 1 se puede identificar que en la parte superior se encuentran los “Efectos”, y en la parte inferior las “Causas”, de la existencia del problema principal.

Entre las “Causas” que hacen que los procesos de la Técnica de Aprendizaje Cooperativo de Rompecabezas en un salón de clase se tornen laboriosos, se detallan las siguientes:

- Se requiere la gestión de un ambiente físico adecuado para el número de alumnos que participan en la sesión. Esto puede generar dificultad para realizar reuniones grupales, ya que se puede presentar diversos motivos de retraso como diferencias en tiempo o lugar geográfico.
- La realización de las tareas que forman parte de esta técnica puede tomar mucho tiempo, y muchas veces requiere de personal adicional para poder minimizar el impacto de tiempo empleado. Por ejemplo, la repartición de las tareas previas del tema puede tomar mucho tiempo para sesiones con varios grupos de alumnos.
- Se requiere la gestión e impresión de material físico para el desarrollo de la sesión de clase. Esto puede demandar tiempo y coordinaciones previas según los procesos institucionales que maneja la entidad que promueve la ejecución de esta técnica.

Entre los “Efectos” de que los procesos de la Técnica de Aprendizaje Cooperativo de Rompecabezas en un salón de clase se tornen laboriosos, se detallan las siguientes:

- Las actividades logísticas se incrementan, debido a que se deben realizar coordinaciones para gestionar lugares físicos adecuados para el número de integrantes; además se debe gestionar la impresión de material físico para la sesión de clase.
- La realización de las tareas que forman parte de la técnica se vuelve tediosa, debido a que su ejecución requiere de mucha dinámica por parte de los organizadores. Es por ello que muchas veces se llega al extremo de emplear tiempo considerable durante las coordinaciones entre cada una de las tareas de una sesión; y debido a esto al final se dispondrá de un menor tiempo efectivo para la realización de la técnica.

- Se presentan problemas al medir la participación de los alumnos, debido a que al manejar un número considerable de alumnos, el Docente no puede darse abasto para atender o inculcar de igual manera la participación de todos los alumnos dentro de la sesión; lo cual lleva a que los alumnos no puedan aprovechar completamente los beneficios que presenta esta técnica.

Es por ello que, debido a los puntos anteriormente descritos, surgió la necesidad de poder simplificar los procesos de implementación de la Técnica de Aprendizaje Cooperativo de Rompecabezas dentro de un salón de clase. En respuesta a esta necesidad, la presente tesis presentó una propuesta para disminuir el tiempo empleado para la realización de las tareas que forman parte de esta técnica; mitigar la gestión e impresión de material físico debido a que todo material se encontrará disponible de manera virtual; y proponer el empleo de un lugar virtual en vez de uno físico para el desarrollo de actividades grupales. Esta solución se basó en el desarrollo de un Sistema Web enfocado en la enseñanza de temas de desarrollo de software, específicamente a la elaboración cooperativa de Casos de Uso.

1.1.3. Objetivo General

El objetivo del presente proyecto es el de analizar, desarrollar e implementar un Sistema Web que permita simplificar los procesos de implementación de la Técnica de Aprendizaje Cooperativo de Rompecabezas; permitiendo de esta manera su aplicación en la enseñanza de Desarrollo de Software, enfocándose específicamente en la elaboración cooperativa de Casos de Uso.

1.1.4. Objetivos Específicos

Para realizar la evaluación de los objetivos específicos de la presente tesis, se decidió emplear la Técnica de Árbol de Objetivos.

La Técnica de Árbol de Objetivos viene a ser la versión positiva de la Técnica de Árbol de Problemas. Esta permite determinar las áreas de intervención que plantea el proyecto. Para su elaboración se parte del Árbol de Problemas. Es necesario revisar cada problema (negativo) y convertirlo en un objetivo (positivo) realista y deseable. Así, las causas se convierten en medios y los efectos en fines. A continuación en la Figura 2, se presenta el Árbol de Objetivos de la presente Tesis:

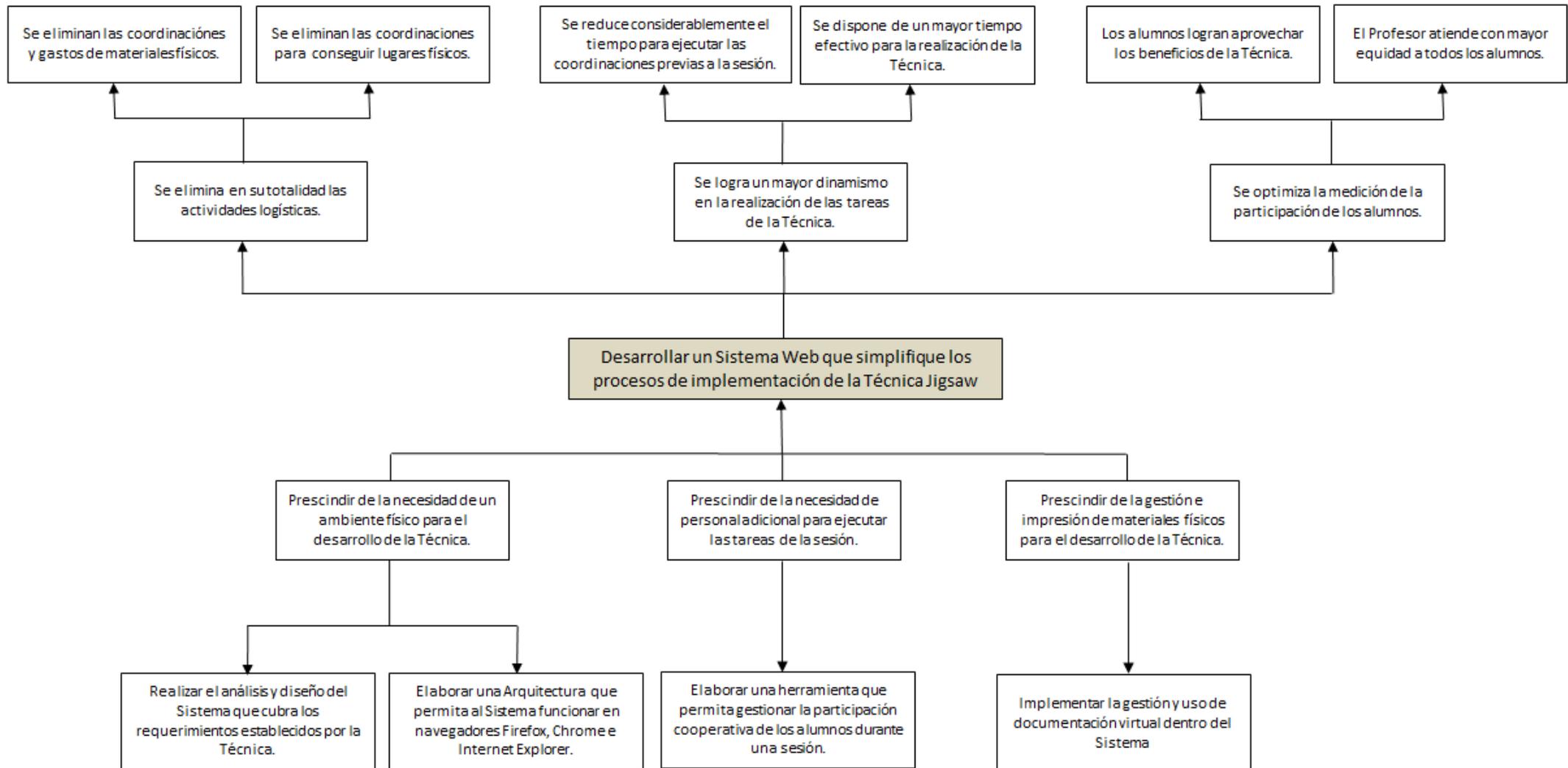


Figura 2: Árbol de Objetivos de la presente Tesis

Del Árbol de Objetivos mostrado en la Figura 2 se puede identificar que en la parte superior se encuentran los “Fines” del objetivo principal. Los “Fines” son la representación positiva de los “Efectos” del problema principal presentado en el Árbol de Problemas de la Figura 1.

En la parte inferior del Árbol de Objetivos se encuentran los “Medios” para poder lograr que se cumpla el objetivo principal. Los “Medios” buscan dar solución a las “Causas” presentadas en el Árbol de Problemas de la Figura 1. Es a partir de estos “Medios” que se permite identificar más detalladamente los objetivos específicos del objetivo principal.

Es por ello que, según el Árbol de Objetivos mostrado en la Figura 2, los objetivos específicos de la presente tesis son:

- Elaborar el análisis y diseño del Sistema que cubra los requerimientos establecidos por la Técnica de Aprendizaje Cooperativo de Rompecabezas.
- Elaborar una Arquitectura que permita al Sistema funcionar en navegadores Web (específicamente: Firefox, Google Chrome e Internet Explorer).
- Elaborar una herramienta que permita gestionar la participación cooperativa de los alumnos durante una sesión de la Técnica de Aprendizaje Cooperativo de Rompecabezas.
- Implementar la gestión y uso de documentación virtual dentro del Sistema.

1.1.5. Resultados Esperados

Luego de cumplir con los cuatro objetivos específicos planteados en la sección anterior por la presente tesis, se obtuvieron los siguientes resultados:

- Se presenta un Sistema Web compatible con los navegadores web Mozilla Firefox, Google Chrome e Internet Explorer, que brinda funcionalidades que dan soporte a las fases que comprende una sesión de clase de la Técnica de Aprendizaje Cooperativo de Rompecabezas. Estas funcionalidades se dividen en los módulos de: Planificación, Ejecución y Evaluación.
 - El módulo de Planificación (fase 1 y 2 de la Técnica de Aprendizaje Cooperativo de Rompecabezas) se encarga de la creación de una

Dinámica y la configuración de las etapas en las que participarán los alumnos.

- El módulo de Ejecución (fase 3 y 4 de la Técnica de Aprendizaje Cooperativo de Rompecabezas) se encarga de realizar las sesiones cooperativas entre los alumnos en un ambiente virtual.
- El módulo de Evaluación (fase 5 y 6 de la Técnica de Aprendizaje Cooperativo de Rompecabezas) se encarga de la creación de una evaluación relacionada a una Dinámica, la realización de la evaluación por parte de los Alumnos, y finalmente la calificación de dichas evaluaciones por parte del Docente.
- El Sistema Web presenta una herramienta denominada Espacio Cooperativo que permite la elaboración de los Casos de Uso para las etapas de Grupos *Jigsaw* y Grupos Expertos:
 - Esta herramienta permite la edición de texto de manera concurrente por cada uno de los integrantes que participan de la sesión.
 - Esta herramienta presenta de manera visual la lista de los integrantes que participan de la sesión.
 - Esta herramienta presentará un chat para la comunicación y coordinación entre los integrantes que participan de la sesión.
- El Sistema Web presenta funcionalidades de gestión de documentación virtual:
 - El Docente puede realizar la asignación de documentación complementaria a la configuración de una Dinámica, por medio de adjuntar archivos.
 - Esta documentación está disponible en el Sistema y se permite su descarga o visualización de contenidos.

1.2. Marco conceptual del problema

En esta sección se describirán los conceptos y temas que se han tomado en cuenta para estructurar el desarrollo del Sistema Web *Jigsaw*. Entre los temas a detallar se encuentran: los conceptos sobre aprendizaje cooperativo y la

descripción de algunas técnicas existentes; y finalmente se detallará el empleo de la Técnica de Aprendizaje Cooperativo de Rompecabezas dentro en un salón de clase.

1.2.1. Aprendizaje Cooperativo

El aprendizaje Cooperativo está basado en la cooperación de los estudiantes para lograr un objetivo común [6]. En este proceso el estudiante no sólo está a cargo de su aprendizaje, sino del aprendizaje de todos sus compañeros de grupo, logrando así desarrollar habilidades de comunicación, argumentación y discusión.

a) Características de un Aprendizaje Cooperativo

- Es una estructura (Técnica o Método) predefinida de interacción y libre contenido.
- Es un enfoque muy centrado o controlado por el docente.
- Los individuos son orientados por una estructura que permite el logro de una meta o producto final específico, generalmente un contenido.
- Requiere un bajo grado de conocimiento de los estudiantes para trabajar en grupo. Es recomendado para enseñar a los estudiantes los procesos básicos de interacción social y el conocimiento de temas de carácter fundamental.
- Hace énfasis en los procesos motivacionales, definidos por el método o la actividad.
- Hace énfasis en los pasos y/o tareas específicas que se deben desarrollar en la actividad [11].

b) Otras Técnicas de Aprendizaje Cooperativo

A continuación se describirán otras técnicas de aprendizaje cooperativo existentes:

Computational Analysis of Knowledge Sharing

Es un ambiente CSCL (*Computer-Supported Collaborative Learning*) que permite a grupos de estudiantes solucionar cooperativamente problemas que involucren

el concepto de orientación a objetos utilizando la técnica OMT (*Object Modeling Technique* propuesta por Jim Rumbaugh). Este sistema cuenta con un espacio de trabajo compartido y una interfaz de comunicación donde los estudiantes pueden construir en común un diagrama, mientras que están trabajando cooperativamente, existe un motor de análisis funcionando dinámicamente, determinando la interacción y recomendando acciones a un agente que idealmente se encuentra en línea supervisando a los estudiantes mientras que están aprendiendo. Este trabajo es importante porque muestra una forma estructurada de comunicación, que es el eje principal de la investigación [12] [13] [14].

The Collaborative Multi-User Editor Project IRIS

En esta investigación se identifica y se satisface las necesidades reales para la edición cooperativa en red. Además se propone un ambiente flexible e integrable para la edición en grupo llamado IRIS, el cual presenta ventajas como: puede usarse en redes de banda ancha, soporta los puestos de trabajo desconectados así como los puestos de trabajo móviles con bajo ancho de banda para establecer enlaces de comunicación (El usuario o la aplicación de interfaz de usuario puede determinar cuándo distribuir las actualizaciones y cuándo recibir las actualizaciones), soporta la colaboración síncrona y asíncrona, soporta la edición cooperativa de gráficos y finalmente, permite la importación y exportación de herramientas [15].

1.2.2. Técnica de Aprendizaje Cooperativo: *Jigsaw*

La Técnica de Aprendizaje Cooperativo de Rompecabezas (*en inglés: Jigsaw Teaching Technique*) es un método que intenta proveer un Ambiente de estudio de colaboración. Su desarrollo comenzó en los años 70 con E. Aronson en la Universidad de Texas y en la Universidad de California y desde entonces cientos de escuelas han basado sus actividades de clase en dicha técnica con mucho éxito [2].

La técnica es desarrollada con interacción personal, sin apoyo tecnológico. En general, el aprendizaje cooperativo o *Collaborative Learning* (CL) es complejo de implementar. Las investigaciones sobre CL indican que existen muchas metodologías y diferente perspectivas teóricas [10].

En el aprendizaje cooperativo, los pequeños grupos de estudiantes trabajan juntos para maximizar su aprendizaje como grupo y así también el aprendizaje individual [9]. Este tipo de actividades encaja en las estrategias de aprendizaje “constructivista”. Esto significa un cambio paradigmático respecto a la tradicional educación conductista.

La tecnología podría proporcionar las herramientas para mejorar la comunicación y la administración de la información; lo cual podría tener un impacto positivo en los procesos educativos. Esto debe cambiar las actitudes respecto al conocimiento y la forma del pensamiento [11]. Sin embargo, hay desafíos al intentar introducir las computadoras para apoyar el CL en los salones de clase. Una de estas actitudes es hacer que el profesor acepte su papel de guía o de asesor cognoscitivo para este nuevo proceso aprendizaje de una manera natural. Otro desafío es desarrollar las herramientas convenientes del CL.

1.3. Plan del Proyecto

1.3.1. Métodos y procedimientos

Para el desarrollo de la presente tesis se empleó la metodología RUP (*Rational Unified Process*). A continuación una breve descripción de esta metodología y la justificación de su uso.

Descripción

El Proceso Unificado Rational o RUP (*Rational Unified Process*), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, que se constituye como una de las metodologías más extendidas y conocidas por su amplia difusión comercial [16].

El RUP es un producto creado por Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso) [16].

Por qué se eligió RUP

La presente tesis buscó realizar la implementación de un software en un mediano plazo. Por lo tanto, para poder administrar todo el desarrollo del software se requirió de una metodología que permita ordenar las etapas del proyecto, mantener una documentación básica para la persistencia y garantizar el manejo de todas las variables que afectarán a un proyecto.

RUP es una metodología diseñada específicamente para desarrollo de software. El ciclo de vida que propone es ordenado en cuanto a sus fases y por ser iterativo e incremental, permitirá un afinamiento del producto cada vez más preciso. Considera también el manejo de requisitos y todo el proceso depende de ellos, esto permite que al final los usuarios tengan lo que necesitan. Por su estructura permite una buena trazabilidad de los requisitos, hace fácil el mantenimiento y posibles modificaciones.

RUP y UML están ligados, por tanto la presencia de la visualización de los procesos a manejar hará más fácil el trabajo y permite guardar las ideas plasmadas en diagramas que cualquier analista diseñador podrá entender. RUP, además, induce a realizar controles sobre el proyecto como son la Gestión de Configuración, Gestión de Riesgos, entre otros.

Fases del desarrollo de software según RUP

La metodología RUP divide el desarrollo del software en cuatro fases [16]:

- **Concepción:** En esta fase se definirá la visión, el alcance del proyecto en función al análisis de los requerimientos del sistema. Se elaborarán los siguientes artefactos: Documento de visión, Catálogo de requisitos, Especificación de requerimientos de software y Documento de Clases de Análisis
- **Elaboración:** En esta fase se definirá el diseño y la arquitectura óptima para el desarrollo del sistema. Se elaborarán los siguientes artefactos: Documento de arquitectura, Prototipo de la solución y Plan de pruebas unitarias y de integración.
- **Construcción:** En esta fase se construirá el sistema y se realizarán las pruebas necesarias para garantizar su correcto funcionamiento y el

cumplimiento de los requerimientos. Se construirán las versiones 1.0, Beta y Final

- **Transición:** Se pondrá a disposición del usuario la versión final del sistema de información para la realización de las pruebas finales.

El desarrollo del Sistema Web *Jigsaw* se llevó a cabo mediante la ejecución de las cuatro etapas arriba mencionadas, las cuales fueron divididas en iteraciones. A continuación en el cuadro 1 se muestra un detalle de lo realizado en cada etapa, así como el número de iteraciones realizadas.

ETAPA	DESCRIPCIÓN	ITERACIONES
Concepción	En esta etapa se realizó el levantamiento de información y la investigación necesaria para la solución del problema, así como un análisis e identificación de requisitos con lo cual se procederá a modelar el negocio, un avance preliminar de la especificación de requisitos de software (ERS)	1
Elaboración	En esta etapa se culminó la especificación de requisitos de software (ERS) en base a los resultados obtenidos en la fase de Concepción. Además se inició el análisis y diseño del sistema, presentación de prototipo de solución y se definió la arquitectura de software que se utilizará en el desarrollo del mismo.	1
Construcción	En esta etapa se realizó el diseño y la programación del Sistema. Además se realizó el plan de pruebas.	5
Transición	Esta etapa no se desarrolló en la presente Tesis.	0

Cuadro 1: Descripción de Etapas e Iteraciones del Sistema Web Jigsaw

1.4. Estado del Arte

La elaboración de una Sesión de Clase empleando la Técnica de Aprendizaje Cooperativo de Rompecabezas presenta un grupo de necesidades que la presente tesis agrupa en tres módulos: el Módulo de Planificación (que permite configurar la creación de la sesión), el Módulo de Ejecución (que permite la interacción entre los alumnos) y el Módulo de Evaluación (que permite rendir y corregir una prueba). En esta sección se presentará la descripción las funcionalidades de tres aplicaciones que individualmente logran cubrir un grupo de necesidades de cada módulo. Luego se procederá a realizar una evaluación mediante un cuadro comparativo sobre el empleo de estas tres aplicaciones, en donde se podrá concluir la necesidad de unificar dentro de un solo sistema las funcionalidades de cada una de las aplicaciones presentadas. Esto nos permitirá establecer de manera general los requerimientos del Sistema Web *Jigsaw*. Finalmente se detallarán los supuestos que este Sistema va a presentar.

1.4.1. Aplicaciones existentes

En esta sección se describirá que aplicaciones existentes son utilizadas actualmente para resolver parte del problema propuesto.

A. GOOGLE DOCS

Google Docs es un conjunto de programas gratuitos online que se emplean para crear documentos y ofrece la posibilidad de realizar la colaboración en grupo. Incluye un procesador de textos, una hoja de cálculo, un programa de presentación de diapositivas y un editor de formularios destinado a crear encuestas [19].

Descripción de la Herramienta

Esta herramienta se puede considerar como un gestor online de documentos, ya que permite subir todo tipo de documentos y desde allí ordenarlos en carpetas y crear enlaces para compartirlos con cualquier persona o página web. Esta herramienta presenta las siguientes funcionalidades:

- Convertir la mayoría de tipos de archivo al formato de Google Docs.
- Darle estilo y formato a los documentos con opciones como pintar formato, márgenes, espaciado y fuentes.

- Invitar a otros usuarios a colaborar en un documento y permitirles ver, modificar o comentar.
- Colaborar online en tiempo real y chatear con otros colaboradores.
- Traducir un documento a otro idioma.
- Enviar documentos como archivos adjuntos a otras personas.

Uno de los mayores atractivos de Google Docs es el poder compartir documentos con otros usuarios, colaborando en su creación y edición hasta llegar a una versión final y así poder publicarlo con una dirección de internet propia.

Esta herramienta presenta las siguientes opciones para compartir archivos:

- Público en la web: Se crea un enlace con una dirección de internet que se puede dar a cualquiera para que acceda al documento. Además el documento puede ser encontrado por buscadores como Google.
- Cualquier usuario que reciba el enlace: El documento es accesible por cualquiera que reciba el enlace pero no se puede encontrar desde los buscadores web. Para esta opción y la anterior se puede indicar el tipo de permiso que se tiene, es decir, si los que accedan al documento tan solo pueden ver el documento o si también pueden editarlo.
- Privado: Es la opción por defecto. No se comparte con nadie.

B. Herramienta desarrollada por la Universidad de Chile

En referencia a los inconvenientes en la administración y ejecución de ambientes de aprendizaje cooperativo, la Universidad de Chile ha desarrollado y puesto en funcionamiento una herramienta de software que proporciona el soporte para el diseño y desarrollo de un ambiente de aprendizaje basada en la Técnica de Aprendizaje Cooperativo de Rompecabezas [10].

Descripción de la Herramienta

Esta herramienta de software se ha enfocado en la edición cooperativa de texto. Su interfaz gráfica presenta una herramienta de la charla (Chat) que permite que los participantes se comuniquen durante el proceso de edición cooperativa.

A continuación en la Figura 3 se presenta el detalle de la interfaz gráfica de esta herramienta:

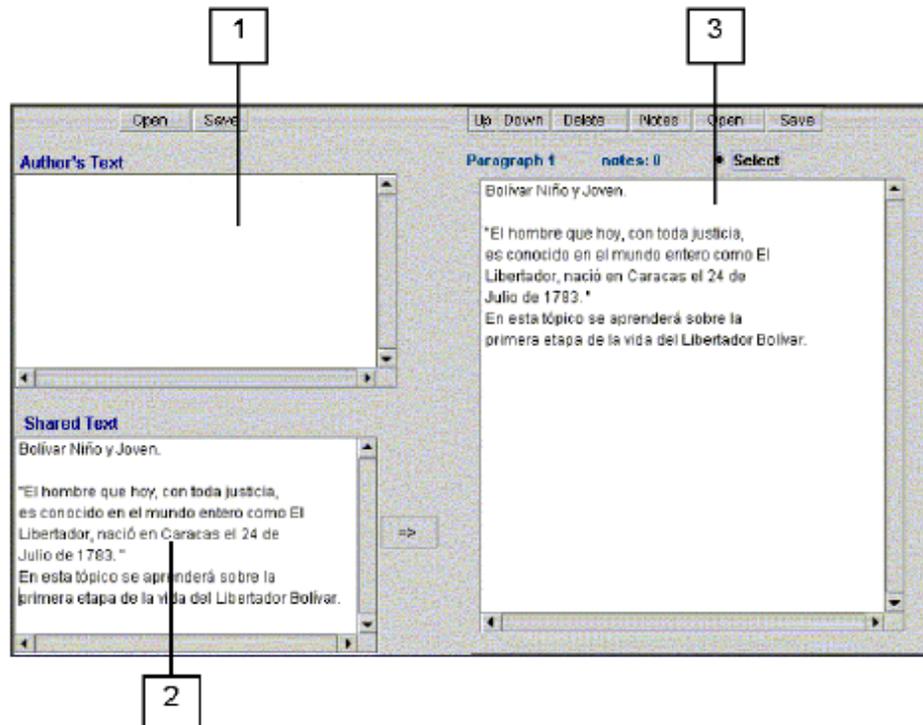


Figura 3: Interfaz de Editor Cooperativo [10]

De la Figura 3 se puede observar que esta herramienta presenta un editor de colaboración con tres secciones principales:

- Espacio de trabajo privado: este espacio permite el almacenar trabajo y notas personales. Esta sección no puede ser accesible a los otros participantes (ver número 1).
- Espacio de trabajo público: el usuario lo utiliza para colocar información disponible para el grupo (ver número 2).
- Espacio de trabajo compartido: este espacio muestra la contribución del todo el grupo. Incluye párrafos con las cualidades del colaborador (autor, fecha de la creación, notas asociadas, etc.) (Véase el número 3).

Otras de las funcionalidades que se presenta es que los docentes pueden preparar las actividades de cooperación basadas en soporte tecnológico con esta herramienta. El propósito es que ellos lo logren en una secuencia de cinco fases.

- Autenticación y registro de participantes.
- Definición del contenido de la Sesión.
- Definición de la actividad.
- Especificación de la evaluación.

Al usar el sistema, el profesor debe abrir una sesión y elegir la sesión en la cual se desea trabajar, o crearla si no existe todavía. A continuación el profesor tendrá acceso a un escritorio del trabajo con estas herramientas disponibles: cuartos de la conversación (Chat), un editor cooperativo y herramientas de la configuración del sistema.

C. Campus Virtual PUCP

Para el manejo de la información académica de sus alumnos e información institucional, la Pontificia Universidad Católica del Perú, por medio de su página web, pone a disposición de su comunidad una intranet denominada “Campus Virtual PUCP”.

Descripción de la Herramienta

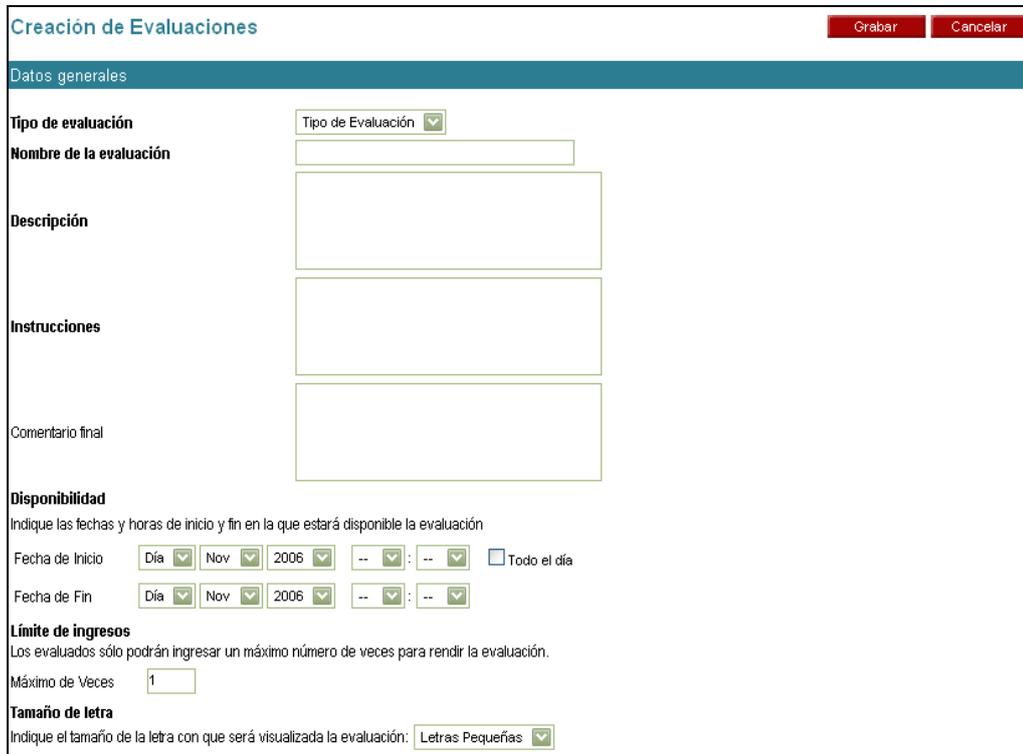
La funcionalidad principal de este Sistema Web se dirige a brindar información detallada e histórica del desempeño académico de cada alumno que inicia sesión. Se presenta al Usuario un sitio virtual con la información personalizada asignada durante los ciclos académicos; además de brindar la opción de poder realizar evaluaciones y encuestas en línea según sea requerido por los docentes o la Institución [20].

El módulo de Evaluaciones y encuestas permite a los docentes la creación y programación de estas; mientras que a los alumnos les permite el poder rendirlas en la fecha y duración asignadas. Este módulo cuenta con tres opciones principales que pueden ser empleadas en el proceso de creación:

- Evaluaciones.
- Preguntas.
- Plantillas.

Para la creación de Evaluaciones y encuestas se registran el nombre, descripción, instrucciones y las características que la pregunta o evaluación

tendrán. A continuación en las Figuras 4 y 5, se muestra un ejemplo de una pantalla de creación.



Creación de Evaluaciones Grabar Cancelar

Datos generales

Tipo de evaluación:

Nombre de la evaluación:

Descripción:

Instrucciones:

Comentario final:

Disponibilidad
Indique las fechas y horas de inicio y fin en la que estará disponible la evaluación

Fecha de Inicio: Día Nov 2006 : -- : -- Todo el día

Fecha de Fin: Día Nov 2006 : -- : --

Límite de ingresos
Los evaluados sólo podrán ingresar un máximo número de veces para rendir la evaluación.

Máximo de Veces:

Tamaño de letra
Indique el tamaño de la letra con que será visualizada la evaluación:

Figura 4: Creación de Evaluaciones y Encuestas en el Campus Virtual PUCP I [20]



Opciones avanzadas

Seleccione la opciones que desee:

- Calificación en línea**
Es calificada en el momento de rendir la evaluación. Opción válida sólo para evaluaciones.
- Mostrar la evaluación pregunta por pregunta**
Sino selecciona esta opción, la evaluación se mostrará en una sola página.
- Mostrar resultados al detalle?**
Muestra a los evaluados los resultados de cada pregunta.
- Mostrar la respuesta correcta?**
Muestra a los evaluados la respuesta correcta de cada pregunta.
- Mostrar preguntas al azar?**
Muestra las preguntas en forma aleatoria.
- Permitir al evaluado modificar sus respuestas?**
Permite al evaluado modificar sus respuestas después de contestadas.
- Permitir al evaluado retroceder las páginas?**
Permite al evaluado retroceder las páginas al rendir una evaluación.
Esta opción no es compatible con la opción de preguntas al azar y la opción de mostrar la evaluación pregunta por pregunta.
- Incluir las imágenes de las preguntas en la misma página**
Permite visualizar las imágenes de las preguntas en la misma página de la evaluación.
Debe tener en cuenta que las imágenes no deben ser muy grandes.
- Respuestas anónimas**
Permite registrar las respuestas en forma anónima. Opción válida sólo para encuestas.
- Integrar la evaluación al sistema de tareas**
Permite asociar la evaluación con una tarea. El evaluador podrá analizar al evaluado junto con las demás tareas que se le han asignado.
- Mostrar preguntas por niveles?**
Los evaluados deberán pasar un nivel de preguntas para poder continuar con las siguientes.
Número de Niveles
- Duración de la Evaluación.**
Muestra a los evaluados un reloj y el tiempo límite de la evaluación.
Horas : Minutos
- Control de tiempo**
Considera el tiempo que le va quedando al evaluado cada vez que ingresa a rendir la evaluación.

Figura 5: Creación de Evaluaciones y Encuestas en el Campus Virtual PUCP II [20]

Para la creación de Preguntas que formarán parte de la Evaluación, el Sistema presenta las opciones de creación de nueva pregunta y la búsqueda de preguntas existentes en el banco de preguntas que posee el curso. La creación de preguntas se da asignándole tipos predefinidos: lógicas, de opción múltiple, de respuesta múltiple, de ordenamiento, de llenado de espacio, respuesta escrita y unión de columnas.

Las plantillas se crean como ayuda a los tipos de Evaluación o Encuesta de respuesta múltiple y opción múltiple: estas brindan la opción de registrar respuestas predefinidas que se pueden asociar a cualquier pregunta de estos tipos.

Una vez concluido el proceso de creación y programación de la evaluación o encuesta, los alumnos podrán tener acceso a estas por medio de su sesión al Campus Virtual PUCP.

En la Figura 6, se muestra un ejemplo de la vista de las preguntas que se deberán responder. En general se responden las preguntas y graban las respuestas.

Vista preliminar de la pregunta
Cerrar

Pregunta Relacione las columnas

Seleccione las respuestas más adecuadas a las definiciones, relacionadas a polimorfismo, que se muestran en la segunda columna

<ol style="list-style-type: none"> 1. Sobreescritura virtual 2. Ocultamiento 3. Sobrecarga 	<p>A. <input type="checkbox"/> 1</p> <p>B. <input type="checkbox"/> 3</p> <p>C. <input type="checkbox"/> 2</p>	<p>Método con mismos parámetros y valor de retorno en una clase base y en la derivada y que se puede implementar en Java</p> <p>Clase que contiene dos o más métodos con el mismo nombre, pero tienen diferentes argumentos</p> <p>Método con mismos parámetros y valor de retorno en una clase base y en la derivada y que NO se puede implementar en Java</p>
---	--	---

Pregunta Ordene las alternativas

Indique la secuencia en que se deben desarrollar las siguientes actividades de especificación de requerimientos.

1. Identificar casos de uso
2. Identificar actores
3. Especificar flujos alternativos
4. Determinar catálogo de requisitos
5. Especificar flujo básico
6. Determinar relaciones Include y Extend.

Respuestas: ordene de acuerdo al enunciado

1°	4. Determinar catálogo de requisitos	<input type="checkbox"/>
2°	-	<input type="checkbox"/>
3°	-	<input type="checkbox"/>
4°	-	<input type="checkbox"/>
5°	-	<input type="checkbox"/>
6°	-	<input type="checkbox"/>

Pregunta Elija la(s) alternativa(s)

Seleccione los procesos de Métrica Versión 3 que contienen actividades equivalentes a la disciplina de Requerimientos de RUP

- A. EVS
- B. ASI
- C. DSI

Figura 6: Ejemplo de Evaluación en el Campus Virtual de la PUCP

1.4.2. Evaluación de requerimientos Sistema Web Jigsaw

En esta sección se realizará una evaluación de las funcionalidades de las Herramientas descritas en el punto 1.3.1, las cuales cubren individualmente requerimientos solicitados por la Técnica de Aprendizaje Cooperativo de Rompecabezas. Para efectos de esta evaluación, se tomarán en consideración las siguientes funcionalidades sugeridos por esta técnica:

- Creación de Sesión de Clase.
- Creación de Evaluación.
- Desarrollo de Evaluación.
- Calificación de Evaluaciones.
- Edición Cooperativa de Contenido.
- Comunicación entre Integrantes.
- Gestión de Documentos.
- Servicio de Mensajería.
- Foros de Discusión.

Estas funcionalidades forman parte de los tres módulos propuestos por la presente tesis, las cuales se agruparán de la siguiente manera:

MÓDULO DE PLANIFICACIÓN

- Creación de Sesión de Clase.

MÓDULO DE EJECUCIÓN

- Edición Cooperativa de Contenido.
- Comunicación entre Integrantes.
- Gestión de Documentos.
- Servicio de Mensajería interna.
- Foros de Discusión.

MÓDULO DE EVALUACIÓN:

- Creación de Evaluación.
- Desarrollo de Evaluación.
- Calificación de Evaluaciones.

A continuación en el Cuadro 2, se realiza una comparación de las funcionalidades de cada una de las herramientas presentadas en la sección 1.3.1, contra las funcionalidades listadas en esta sección:

	Creación de Sesión de Clase	Creación de Evaluación	Desarrollo de Evaluación	Calificación de Evaluaciones	Edición Cooperativa de Contenido	Comunicación entre Integrantes	Gestión de Documentos	Servicio de Mensajería	Foros de Discusión
Google Docs	✗	✗	✗	✗	✓	✓	✓	✗	✗
Herramienta Univ. Chile	✓	✗	✗	✗	✓	✓	✗	✗	✗
Campus Virtual PUCP	✗	✓	✓	✓	✗	✗	✗	✗	✗

Cuadro 2: Evaluación de Funcionalidades de la Técnica de Aprendizaje Cooperativo de Rompecabezas

Del cuadro presentado se puede observar que:

- Si bien Google Docs es una herramienta muy completa en gestión y edición cooperativa de documentos, su funcionalidad solo abarca un módulo (el de Ejecución) de los tres planteados por la presente tesis. Esta herramienta solo cubre tres de las cinco funcionalidades planteadas por este módulo.
- La herramienta desarrollada por la Universidad de Chile, presenta funcionalidades que abarcan dos módulos (el de Planificación y Ejecución) de los tres planteados por la presente tesis. Esta herramienta cubre en su totalidad la funcionalidad del módulo de Planificación, mientras que en el módulo de Ejecución solo cubre dos de las cinco funcionalidades planteadas.
- El Campus Virtual PUCP, presenta funcionalidades que abarcan un módulo (el de Evaluación) de los tres planteados por la presente tesis. Esta herramienta cubre en su totalidad las tres funcionalidades planteadas por este módulo.

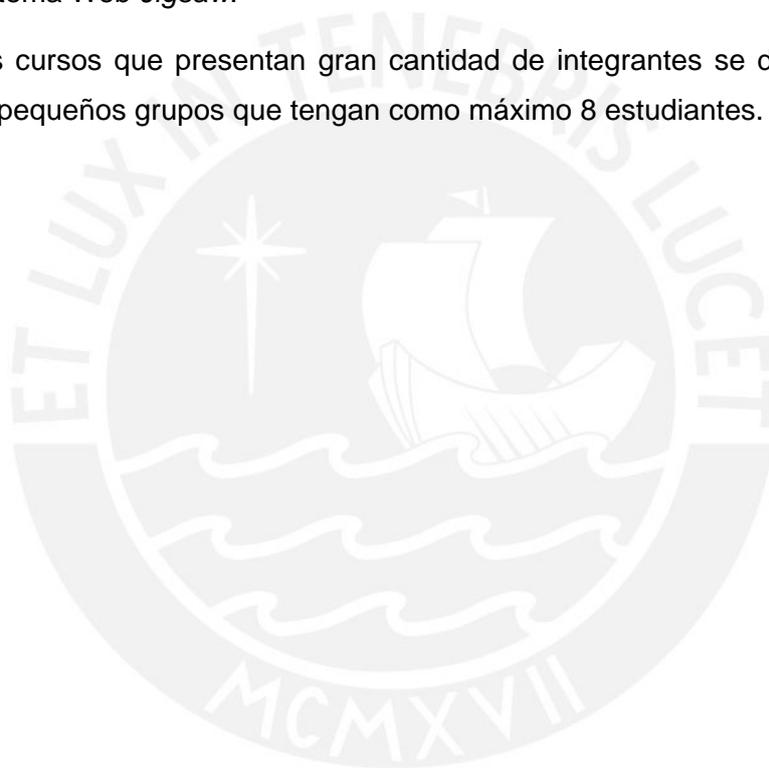
Según lo detallado en las observaciones anteriores, se puede ver que estas tres herramientas aportan, pero de manera aislada, funcionalidades identificadas en los módulos planteados por la presente tesis. Según su descripción e implementación, estas funcionalidades podrían también contribuir a simplificar los procesos de implementación de la Técnica de Aprendizaje Cooperativo de Rompecabezas.

Es por ello que se propone la necesidad de creación de un Sistema Web (Sistema Web *Jigsaw*) que tenga como objetivo el integrar las funcionalidades de cada una de estas herramientas dentro de un solo ambiente web; el cual permitirá aportar los procedimientos para simplificar de manera conjunta las tareas y funcionalidades requeridas para implementar esta técnica de enseñanza cooperativa.

1.4.3. Requisitos del Sistema Web Jigsaw

En esta sección se detallará el funcionamiento mínimo que se deberá cumplir con respecto al Sistema Web *Jigsaw* y los grupos de una Sesión de Clase empleando la Técnica de Aprendizaje Cooperativo de Rompecabezas:

- El Usuario solo tiene acceso a la interfaz cooperativa dentro del Sistema Web *Jigsaw*, cuando se estén realizando las etapas asociadas (“Grupo Experto” y “Grupo *Jigsaw*”) a este, dentro de la Sesión de Clase empleando la Técnica de Aprendizaje Cooperativo de Rompecabezas.
- El Usuario podrá ingresar a su sesión cooperativa cuantas veces lo crea conveniente; sin que esto afecte el desempeño del Sistema Web *Jigsaw* y en la Sesión de Clase a la que ha sido asignado.
- Todos los Usuarios que forman parte de una sesión cooperativa pueden trabajar sobre un mismo documento sin que esto afecte el desempeño del Sistema Web *Jigsaw*.
- Los cursos que presentan gran cantidad de integrantes se deben distribuir en pequeños grupos que tengan como máximo 8 estudiantes.



Capítulo II: Análisis

En este capítulo, se presentan las soluciones sobre tecnología y software que se emplearon para el desarrollo de la presente tesis. Finalmente se presentará un análisis a alto nivel del Sistema Web *Jigsaw* mediante su Diagrama de Clases, el catálogo de actores que intervienen, y la presentación de los Casos de Uso más importantes.

2.1. Identificación de requerimientos

En esta sección se presentarán los requerimientos funcionales por los módulos de Planificación, Ejecución y Evaluación del Sistema Web *Jigsaw*. A cada uno de estos requerimientos se les ha asignado un nivel de prioridad, con la finalidad de priorizar todos aquellos que sean los más importantes y contribuyan a lograr los objetivos de la presente Tesis (Prioridad 1); para luego continuar con aquellos requerimientos que puedan darle valor agregado al Sistema (Prioridad 2).

2.2.1. Requerimientos funcionales

Un requerimiento funcional es aquel que define el comportamiento interno de un software. Este describe principalmente los cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que mostrarán lo que debe llevarse a la práctica.

El Sistema Web *Jigsaw* busca cubrir los requerimientos establecidos por la Técnica de Aprendizaje Cooperativo de Rompecabezas para el desarrollo de una Sesión Cooperativa (un objetivo de la presente Tesis), es por ello que estos han sido agrupados en tres módulos: el Módulo de Planificación, el Módulo de Ejecución y el Módulo de Evaluación.

El Módulo de Planificación es administrado y ejecutado por el Docente. En este módulo el Sistema permite al Usuario la creación y configuración de una Dinámica. Esta configuración abarca la creación de Temas Expertos; la gestión de los documentos que se asignarán a la Dinámica; y la asignación manual de alumnos a los Grupos Cooperativos (que se denominarán Grupos *Jigsaw*) y su asignación automática a Grupos Expertos.

El Módulo de Ejecución es ejecutado por los alumnos mediante su participación en las Sesiones Cooperativas de los Grupos *Jigsaw* y Grupos Expertos, que

fueron previamente configurados en el Módulo de Planificación. Este módulo presenta una herramienta que permite la participación cooperativa de los alumnos que participan de una Sesión Cooperativa (un objetivo de la presente Tesis). Esta herramienta permite la edición de contenido de un texto por parte de todos los participantes de dicha sesión; además facilitar la comunicación entre ellos mediante de un servicio de chat. En este módulo el Docente realiza un rol de monitoreo al indicar la disponibilidad de inicio y fin de cada Sesión Cooperativa. Adicionalmente, el Sistema presenta funcionalidades de gestión sobre los documentos virtuales de una Dinámica (un objetivo de la presente Tesis) mediante su descarga o visualización de contenidos.

El Módulo de Evaluación es ejecutado por el Docente mediante la creación de una evaluación mediante el ingreso de enunciados con sus respectivas preguntas y respuestas; y luego realiza la calificación de las respuestas ingresadas por los alumnos de una Dinámica. Este módulo es ejecutado también por los alumnos al rendir la evaluación relacionada a la Dinámica que están participando. Aquí los alumnos registran sus respuestas a las preguntas definidas previamente por el Docente en el módulo de planificación.

En el Cuadro 3, se detallan los requerimientos funcionales para el Módulo de Planificación. Según la información listada, este módulo busca principalmente configurar los contenidos de las posteriores etapas que ejecutará el Alumno.

En el Módulo de Planificación, los requerimientos principales (prioridad 1) se inician mediante la creación de una Dinámica. Una Dinámica parte de la elección de un tema de un curso sobre el cual esta se basará. Una dinámica define Temas Expertos, los cuales funcionarán como "tarea previa" a realizar antes del inicio de la dinámica, y cuya función es la de familiarizar al alumno con los temas que se van a desarrollar dentro la Dinámica. Adicionalmente a esto se puede asignar documentación que permita complementar la comprensión de los conocimientos a ser evaluados. Posteriormente se procede a definir la asignación de los alumnos a cada Grupo Experto y Grupo *Jigsaw*. Finalmente se realiza la creación de una Evaluación referente al tema de la Dinámica, especificando enunciados con sus respectivas preguntas y respuestas.

	MÓDULO DE PLANIFICACIÓN	PRIORIDAD
1	El Sistema permitirá la creación y configuración de una Dinámica en base a un tema de un curso	1
2	El Sistema permitirá la eliminación de una Dinámica antes que esta se inicie.	1
3	El Sistema permitirá el registro de Temas Expertos y su asignación a una Dinámica.	1
4	El Sistema permitirá la eliminación de Temas Expertos asignados a una Dinámica.	1
5	El Sistema permitirá la creación automática de Grupos <i>Jigsaw</i> según el número de Temas Expertos registrados.	2
6	El Sistema permitirá la asignación manual de los alumnos a cada Grupo <i>Jigsaw</i> .	1
7	El Sistema permitirá la asignación de un Tema Experto a cada alumno de un Grupo <i>Jigsaw</i> .	1
8	El Sistema realizará la creación automática de Grupos Expertos.	2
9	El Sistema realizará la asignación automática de alumnos a cada Grupo Experto de acuerdo a su Tema Experto asignado.	2
10	El Sistema permitirá la creación de una evaluación referente al tema de la Dinámica; mediante el ingreso de un enunciado general y la creación de preguntas y con sus respectivas respuestas.	1
11	El Sistema presentará un servicio de mensajería interna entre los integrantes de la Dinámica.	2
12	El Sistema permitirá la asignación de documentos a una Dinámica.	1
13	El Sistema permitirá la eliminación de documentos asignados a una Dinámica.	1

Cuadro 3: Requisitos Funcionales Modulo Planificación

El Módulo de Planificación también presenta requerimientos que brindan un valor agregado a este módulo (prioridad 2), y que buscan principalmente facilitar la experiencia de Usuario frente al Sistema. Estos se encuentran en la generación automática del número de Grupos *Jigsaw* según el número de Temas Expertos registrados; la agrupación automática de alumnos a Grupos Expertos según su posición dentro del cada Grupo *Jigsaw* (por ejemplo: todos los alumnos en la

primera posición de cada grupo formarán el primer Grupo Experto, y así sucesivamente); y la presentación de un servicio de Mensajería que permitirá la comunicación entre los usuarios del Sistema.

En el Cuadro 4, se detallan los requerimientos funcionales para el Módulo de Ejecución. Según la información listada, este módulo se enfoca principalmente en control y realización de las Sesiones Cooperativas de Grupos *Jigsaw* y Grupos Expertos.

En el Módulo de Ejecución, los requerimientos principales (prioridad 1) están conformados por las funcionalidades que presentará el Espacio Cooperativo para la realización de Casos de Uso que se empleara en cada sesión. Este Espacio Cooperativo deberá permitir identificar a los integrantes que forman parte de la sesión; permitir la edición de texto de manera concurrente por parte de cada integrante; y permitir la comunicación y coordinación entre ellos mediante un servicio de chat. Adicionalmente este módulo brinda un menú gráfico al Alumno para participar de una sesión que esté disponible; además de brindar al Docente otro menú gráfico que le permitirá controlar el inicio y final de cada sesión cooperativa.

El Módulo de Ejecución también se presentan requerimientos que brindan un valor agregado a este módulo (prioridad 2), buscando principalmente facilitar la experiencia de Usuario frente al Sistema. Estos se encuentran en la presentación al alumno de un espacio informativo para cada Dinámica en la cual este participa. Dentro de estos espacios se incluye la información del Tema Experto asignado y su documentación de apoyo. El contenido de los menús gráficos presentados al Docente y al Alumno se actualizará en tiempo real, es decir, no será necesario volver a cargar los contenidos de la pantalla para reflejar algún cambio en sus estados. Finalmente, cada Dinámica presentará un Foro de Discusión para el intercambio de ideas y conceptos referentes a los temas propuestos.

Finalmente en el Cuadro 5, se detallan los requerimientos funcionales para el Módulo de Evaluación. Según la información listada, este módulo se enfoca principalmente en la creación del contenido de una evaluación asociada a la Dinámica que se está desarrollando; en el proceso cuando el Alumno rinde dicha Evaluación mediante el registro de sus respuestas; y finalmente la posterior corrección de estas por parte del Docente.

	MODULO DE EJECUCIÓN	PRIORIDAD
1	El Sistema presentará un espacio informativo para el alumno sobre los detalles de la Sesión Cooperativa asignada a una Dinámica creada por el Docente.	2
2	El Sistema presentará al Alumno el detalle de la tarea previa del Tema Experto asignado dentro de la Dinámica.	2
3	El Sistema presentará al Alumno un menú gráfico que indicará la etapa actual de la Dinámica en que participa (Grupo Experto / Grupo <i>Jigsaw</i> / Rendir Evaluación).	1
4	El Sistema presentará al Docente un menú gráfico que le permitirá administrar el inicio y finalización de las etapas de la Dinámica.	1
5	El Sistema actualizará en tiempo real y sin refrescar los contenidos de la pantalla, el estado del Menú gráfico que indicará la etapa actual de la Dinámica en que participa.	2
6	El Sistema actualizará en tiempo real y sin refrescar los contenidos de la pantalla, el estado del Menú gráfico que le permitirá administrar el inicio y finalización de las etapas de la Dinámica.	2
7	El Sistema presentará un foro de discusión por cada Dinámica creada.	2
8	El Sistema presentará un Espacio Cooperativo que permitirá la elaboración de los Casos de Uso para las etapas de Grupos <i>Jigsaw</i> y Grupos Expertos.	1
9	El Espacio Cooperativo para elaboración de los Casos de Uso permitirá la edición de texto de manera concurrente por cada uno de los integrantes que participan de la sesión.	1
10	El Espacio Cooperativo para elaboración de los Casos de Uso permitirá darle formato al texto editado (cambio letra, tamaño de letra, color de letra, subrayado y aplicación de negrilla).	1
11	El Espacio Cooperativo para elaboración de los Casos de Uso mostrará la lista de los integrantes que participan de la sesión.	1
12	El Espacio Cooperativo para elaboración de los Casos de Uso presentará un chat para la comunicación y coordinación entre los integrantes que participan de la sesión.	1
13	El Sistema presentará a los Usuarios los documentos asignados a una Dinámica.	1
14	El Sistema permitirá descargar o ver los contenidos de los documentos asignados a una Dinámica.	1

Cuadro 4: Requisitos Funcionales Modulo Ejecución

En el Módulo de Evaluación, los requerimientos principales (prioridad 1) están conformados por el registro -por parte del Docente- de los enunciados, preguntas y respuestas correspondientes a esta Evaluación. Posteriormente se presenta un espacio virtual que contiene el detalle de las preguntas y respuestas creadas previamente. Este espacio virtual permitirá el registro de una o varias respuestas a una pregunta (dependiendo del tipo es "Opción Simple" u "Opción Múltiple"), o permitirá el adjuntar un archivo de respuesta en el caso de ser una pregunta desarrollada. Adicionalmente el Sistema podrá realizar una corrección automática de todas las preguntas de la Evaluación en caso que no se haya incluido alguna pregunta desarrollada. En caso haya al menos una pregunta desarrollada se procederá a una corrección manual. En este tipo de calificación las preguntas desarrolladas quedarían pendientes de corrección por parte del Docente y las preguntas restantes se encontrarán calificadas por defecto (como en el caso de corrección automática). Para el caso de corrección de preguntas desarrolladas se permitirá la visualización del archivo de respuesta, así como el poder asignar una calificación y comentario a dicha pregunta.

El Módulo de Evaluación también se presentan requerimientos que brindan un valor agregado a este módulo (prioridad 2), buscando principalmente facilitar la experiencia de Usuario frente al Sistema. Estos se encuentran durante la creación de la Evaluación al presentar una vista preliminar de las respuestas asignadas como correctas. Posteriormente durante la corrección de la Evaluación se presenta una vista comparativa de las respuestas registradas por el Alumnos contra las respuestas definidas como correctas dentro de la Evaluación. Además, se presentarán alertas informativas en el caso que el docente tenga evaluaciones por corregir, y el Alumno tenga disponible alguna nota de una evaluación calificada.

	MODULO DE EVALUACIÓN	PRIORIDAD
1	El Sistema presentará un espacio virtual de evaluación donde al alumno se le presentará la lista de preguntas de la evaluación a rendir.	1
2	El espacio virtual de evaluación permitirá al alumno registrar sus respuestas según el tipo de pregunta registrada y opciones disponibles.	1
3	El espacio virtual de evaluación permitirá al alumno el poder adjuntar un archivo de respuesta para el caso de preguntas desarrolladas.	1
4	El Sistema permitirá al Docente la creación de preguntas de Opción Múltiple, Opción Simples y Desarrolladas.	1
5	El Sistema permitirá al Docente la asignación de un puntaje a cada tipo de pregunta creada.	1
6	El Sistema permitirá al Docente el registro del detalle y enunciados para una Evaluación.	1
7	El Sistema permitirá al Docente el registro de las preguntas y enunciados de una evaluación.	1
8	El Sistema permitirá al Docente la visualización preliminar de las respuestas correctas de la evaluación.	2
9	El Sistema permitirá al Docente, durante la corrección de una Evaluación, la visualización de una vista comparativa de las respuestas registradas por el Alumno contra las respuestas correctas de la evaluación.	2
10	El Sistema realizará una corrección automática de una evaluación en caso de tener solamente preguntas de Opción Simple y de Opción Múltiple.	1
11	El Sistema permitirá al Docente la realización de una corrección manual de una Evaluación en caso esta tenga registrada preguntas desarrolladas.	1
12	El Sistema permitirá al Docente ingresar manualmente la calificación de una pregunta desarrollada, además de un comentario referente a la corrección.	1
13	El Sistema mostrará alertas informativas para indicar al Docente las evaluaciones por corregir.	2
14	El Sistema mostrará alertas informativas para indicar al alumno que su evaluación ha sido calificada.	2

Cuadro 5: Requisitos Funcionales Modulo Evaluación

2.2.2. Requerimientos no funcionales

Según su definición, un requerimiento no funcional es una característica requerida del sistema, del proceso de desarrollo, de un servicio prestado o de cualquier otro aspecto del desarrollo que no se satisface añadiendo código, sino cumpliendo con esta como si se tratara de una restricción.

En la sección anterior, los requerimientos funcionales del Sistema Web *Jigsaw* abarcaron tres objetivos (cubrir los requerimientos establecidos por la Técnica de Aprendizaje Cooperativo de Rompecabezas, presentar una herramienta que permita gestionar la participación cooperativa de los alumnos durante una sesión, y gestionar el uso de documentación virtual dentro del Sistema) de los cuatro presentados en la presente Tesis.

El cuarto objetivo de la presente Tesis está conformado por la elaboración de una arquitectura que permita al Sistema el poder funcionar en navegadores web. Este último nos permitirá establecer de manera concreta la mayoría de los requerimientos no funcionales del Sistema Web *Jigsaw*, debido a que nos permite especificar sus restricciones de hardware, software y desempeño. Adicionalmente a estos, también se especificará un requerimiento no funcional sobre el desempeño de procesamiento de datos para los módulos de planificación y ejecución respectivamente.

En el Cuadro 6, se listan los requerimientos no funcionales para el Sistema Web *Jigsaw*. Según la información listada, los requerimientos no funcionales especifican inicialmente la restricción de compatibilidad de funcionamiento a tres navegadores web: Mozilla Firefox, Google Chrome e Internet Explorer.

Se indica además una restricción importante en el uso del Espacio Cooperativo para elaboración de los Casos de Uso: se deberá emplear el mismo tipo de navegador por cada terminal de usuario que participe en la Sesión. Esta restricción se da principalmente porque el Espacio Cooperativo fue desarrollado empleando las librerías del *Open Cooperative Web Framework* v1.0 (el Capítulo IV, sección 4.1.3 presenta mayor información sobre esta librería), el cual ayuda a actualizar los cambios realizados sobre un texto enviando su contenido como un bloque HTML con estilos CSS. Debido a que esta versión de la librería aún no ha desarrollado un filtro de transporte de contenidos multiplataforma, cada uno de estos tres navegadores mencionados pueden interpretar de manera diferente los

estilos CSS, llevando a diferencias en la visualización de los contenidos para cada navegador.

REQUISITOS NO FUNCIONALES	
1	El Sistema será soportado por navegadores Web: Mozilla Firefox, Google Chrome e Internet Explorer.
2	El Espacio Cooperativo para elaboración de los Casos de Uso deberá ser ejecutado empleando el mismo tipo de navegador web en todas sus terminales de usuario durante una sesión cooperativa.
3	El Espacio Cooperativo para elaboración de los Casos de Uso actualizará la información en todas las sesiones involucradas dentro de una Dinámica en un tiempo máximo de 3 segundos.
4	Para el desarrollo del Sistema se empleará el lenguaje de programación Java junto con el framework de desarrollo Struts 2.
5	El Sistema funcionará por medio de un Servidor de Servlets JBoss AS 7.
6	El Sistema empleará de una base de datos MySQL junto a la herramienta de persistencia MyBatis.
7	El Sistema presentará una interfaz de usuario implementada en HTML sin FRAMES, Applets de Java, o animaciones Flash incrustadas.
8	El Sistema hará uso de la Tecnología AJAX por medio de las librerías JQuery y Dojo Toolkit.
9	La interacción con el Sistema se realizará mediante el uso de teclado y ratón.
10	La información de la sesión de un usuario del Sistema, no deberá estar al alcance del público general.

Cuadro 6: Requisitos No Funcionales

Para efectos de desarrollo del Sistema se especifica el empleo del lenguaje Java junto con el *framework* de desarrollo *Struts 2*; además del empleo del motor de base de datos *MySQL* junto a la herramienta de persistencia *MyBatis* (en el Capítulo IV, sección 4.1.3 se brinda un amplio detalle sobre la justificación de uso de estas tecnologías). También se hace una observación importante referente a la programación de las interfaces del Sistema: estas deberán ser hechas empleando HTML junto con las librerías *JQuery* y *Dojo Toolkit*. Estas últimas librerías se emplearán para la creación de animaciones, manejo de tecnología Ajax y consultas al servidor de base de datos (en el Capítulo IV, sección 4.1.3 se brinda un amplio detalle sobre las ventajas que brindan estas

tecnologías). Los temas referentes al mantenimiento de servidores y base de datos escapan al alcance de la presente Tesis, debido a que el alcance abarca hasta la etapa de desarrollo del Sistema.

Adicionalmente se indica la necesidad de privacidad de los datos de los usuarios del Sistema (módulo de planificación); la necesidad que el Espacio Cooperativo actualice en un tiempo máximo de tres minutos los cambios cooperativos realizados a un texto, a todas las terminales de usuario que participan de una sesión (módulo de ejecución); y finalmente se indica que la interacción con el Sistema se realizará únicamente mediante teclado y ratón.

2.2. Análisis de la solución

En esta sección presentará un análisis sobre la realización de las tareas de la Técnica de Aprendizaje Cooperativo de Rompecabezas en un salón de clase. Este análisis nos permitirá sustentar la viabilidad de las propuestas de solución que brindará el Sistema Web *Jigsaw* de acuerdo a los objetivos de la presente Tesis. Finalmente se realizará un análisis tecnológico sobre las tecnologías que fueron seleccionadas para ser empleadas durante el desarrollo.

2.3.1. Evaluación de viabilidad y consideraciones sobre el sistema

La Técnica de Aprendizaje Cooperativo de Rompecabezas ha sido implementada en los ambientes de clase del curso de Sistemas de Información 2 en la Pontificia Universidad Católica Del Perú como refuerzo a las clases magistrales impartidas previamente a cada tema del curso; la cual ha logrado resultados alentadores relacionados al nivel de aprendizaje de los estudiantes y un grado alto de empatía de los alumnos con la técnica.

La ejecución de las tareas que conforman esta técnica fue concebida para ser realizadas dentro de un ambiente físico, requiriendo la distribución de material impreso y presentando la necesidad de asistencia de personal de apoyo [2][17]. Actualmente la ejecución de estas tareas incurre en gastos económicos debido a la necesidad de gestionar la elaboración e impresión de material escrito; además de presentar incertidumbres de tiempo en las coordinaciones para encontrar disponible un ambiente físico adecuado para el número de alumnos participantes.

Es por ello la presente tesis propone el empleo de un ambiente virtual que permitirá cumplir con las tareas requeridas por la Técnica de Aprendizaje Cooperativo de Rompecabezas, además que podrá asegurar la interacción entre sus participantes. Este ambiente virtual será implementado por medio de un Sistema Web que presentará las siguientes ventajas:

- El Sistema Web presentará una alternativa de ahorro en el costo de generación de material escrito para la ejecución de la Técnica de Aprendizaje Cooperativo de Rompecabezas, debido a que ahora este material estará disponible de manera virtual dentro del Sistema, mediante su visualización o descarga.
- El Sistema Web presentará una alternativa de solución a la necesidad de coordinación de ambientes físicos para la ejecución de la Técnica de Aprendizaje Cooperativo de Rompecabezas, debido a que se empleará salas virtuales que permitan la interacción cooperativa entre sus participantes.
- El Sistema Web se presenta como una opción para alentar una participación más activa de los alumnos, debido a que su empleo ya no presenta restricciones geográficas ni de tiempo para participar; salvo la necesidad de contar con una conexión a internet y un navegador web compatible con el Sistema.
- El desarrollo y funcionamiento del Sistema Web se realizará empleando herramientas de código abierto, lo que asegura el evitar gastos de pagos de licencias.

Teniendo en cuenta estas ventajas, el desarrollo del Sistema Web tiene como objetivo cumplir con los siguientes atributos de calidad:

- **Performance:** El tiempo de acceso al sistema no deberá ser mayor a 5 segundos.
- **Disponibilidad:** El sistema estará disponible las 24 horas del día, los 7 días de la semana.

- **Utilidad:** El sistema será apropiado para el uso de dos perfiles (el de Docente y Alumno), de manera que sirvan como soporte de las labores de todas personas que interactúen con él.
- **Usabilidad:** El sistema presentará una interfaz visual y amigable, además proporcionando al usuario mensajes de error y de advertencia.
- **Escalabilidad:** El sistema estará diseñado de tal manera que se puedan agregar nuevas funcionalidades sin que esto afecte de manera crítica el código.
- **Seguridad:** El acceso al sistema estará restringido a usuarios que cuenten con una contraseña.
- **Validación de la información:** La información registrada en el sistema será validada.

2.3.2. Análisis Técnico

Para la elección de las tecnologías que se emplearon para el desarrollo del Sistema Web *Jigsaw* se ha consideraron los siguientes criterios:

- Que las tecnologías a emplear presenten la menor cantidad de dinero a invertir.
- Que las tecnologías a emplear presenten una curva de aprendizaje no muy extensa.
- Que las tecnologías a emplear tengan suficiente documentación disponible y soporte técnico.
- Que el motor de base de datos presente rapidez en la lectura y administración de sus datos.
- Que el servidor de aplicaciones presente facilidad en su configuración, administración y además brinde un alto rendimiento durante su ejecución.
- Que el servidor de aplicaciones presente suficiente documentación disponible y soporte técnico.

Tomando en cuenta estos criterios se procedió a la elección de las siguientes tecnologías (para una mayor información sobre estas tecnologías ver el Capítulo IV, sección 4.3.1):

Framework de Desarrollo Apache Struts 2

La opción inicial para el desarrollo de la presente tesis fue el *Framework Apache Struts* (Struts 1); pero luego de algunas revisiones se optó por el empleo del *Framework Apache Struts 2*. Ambos *frameworks* son *open source* (sin costo de licencia) que emplean el lenguaje de programación Java para su implementación. Las mejoras de Struts 2 frente a Struts 1 se dan en los siguientes puntos:

- *Struts 2* presenta una mayor simplicidad en la manera de comunicarse con una base de datos, a comparación de *Struts 1*.
- *Struts 2* presenta una mayor simplicidad en la codificación de las reglas de negocio de un Sistema, a comparación de *Struts 1* que requiere clases auxiliares.
- *Struts 2* presenta una mayor cantidad de librerías de apoyo para la gestión de interfaces, a comparación de *Struts 1*.

Servidor de Aplicaciones JBoss

JBoss está designado para dar soporte a Sofisticadas Aplicaciones Java que requieran procesos complejos, tales como *Enterprise Java Beans* (EJB) y/o deban tener el potencial para manejar grandes cantidades de procesos. Además, *JBoss* es un programa *open source* (sin costo de licencia), fácil de manejar y administrar, que posee una gran compatibilidad y complemento con la herramienta de mapeo objeto-relacional JPA.

A comparación de otras soluciones tales como *BEA Weblogic* e *IBM Websphere*, que son de software propietario, requieren el pago de licencias y demandan mayor soporte para la realización de sus procesos; *JBoss* se consideró como una opción más confiable y económica.

Base de Datos MySQL

MySQL es un manejador de base de datos *open source* (sin costo de licencia) muy veloz en la realización de operaciones y de gran rendimiento, su bajo consumo lo hace ideal para equipos de escasos recursos. Entre las ventajas que presenta esta base de datos tenemos:

- Presenta un uso más simple para su manejo y administración; además de poseer mayor documentación y soporte técnico en comparación con otras soluciones de software libre (como por ejemplo: *PostgreSQL*).

- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

2.3.3. Definición del sistema

En esta sección se brinda una descripción a alto nivel del sistema. A continuación se presentará el modelo de casos de uso (modelo que muestra la funcionalidad del sistema), el catálogo de actores y finalmente el diagrama de clases de análisis.

A. Modelo de Casos de Uso

Esta sección presenta los diagramas de caso de uso del sistema obtenidos durante el proceso de especificación de requisitos, los cuales permiten mostrar un alto nivel de funcionalidad que el sistema realizará. Inicialmente, se indica el catálogo de actores que interactúa con el sistema y posteriormente la descripción de cada uno de los paquetes con sus respectivos diagramas de casos de uso.

Los Casos de Uso son una técnica para la especificación de requisitos funcionales que actualmente forma parte de la propuesta de UML. Un Caso de Uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que los actores obtienen resultados observables.

A.1. Catálogo de Actores

Los actores son personas u otros sistemas que interactúan con el sistema cuyos requisitos se están describiendo. A continuación en el Cuadro 7, se presenta una descripción de cada uno de los actores que intervienen en el Sistema.

Usuario del Sistema	Cualquier usuario del sistema que posee una cuenta en el Sistema <i>Jigsaw</i> PUCP.
Docente	Docente del Curso, encargado de la planificación de la sesión, seguimiento y evaluación.
Alumno	Participa en la sesión haciendo uso de las herramientas de ejecución.

Cuadro 7: Catálogo de Actores

A.2. Paquetes del sistema

Un Paquete es una agrupación a alto nivel de los Casos de Uso del Sistema. A continuación en la Figura 7, se presentará el diagrama de los paquetes del sistema.

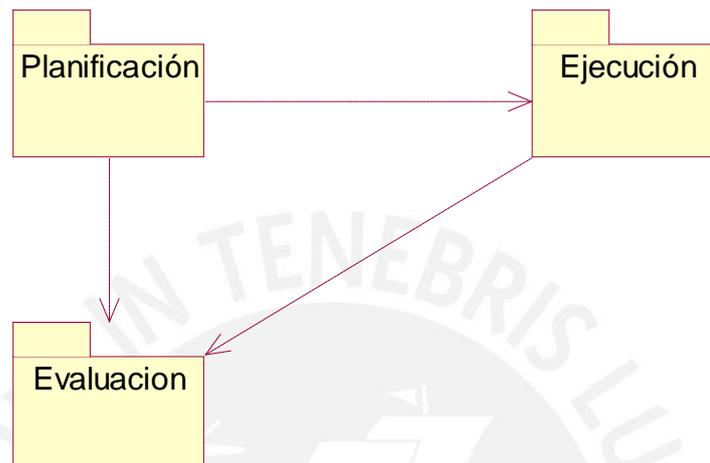


Figura 7: Paquetes del Sistema

En la figura 7 se puede observar la dependencia de los tres paquetes que forman parte del Sistema Web *Jigsaw*. El Módulo de Planificación brinda información a los Módulos de Ejecución y al Módulo de Evaluación. A su vez, el Módulo de Ejecución brinda información al Módulo de Evaluación.

A.3. Casos de Uso por Paquete

A continuación se presentarán los casos de uso de los módulos que conforman el sistema organizados en paquetes.

Paquete Planificación

Este paquete se encarga de la configuración de las dinámicas de un curso aplicando para cada una de ellas la Técnica de Aprendizaje Cooperativo de Rompecabezas. El principal actor de este paquete es el docente ya es el encargado de la configuración de cada sesión.

A continuación en la Figura 8, se presenta el Diagrama de Casos de Uso del Módulo de Planificación.

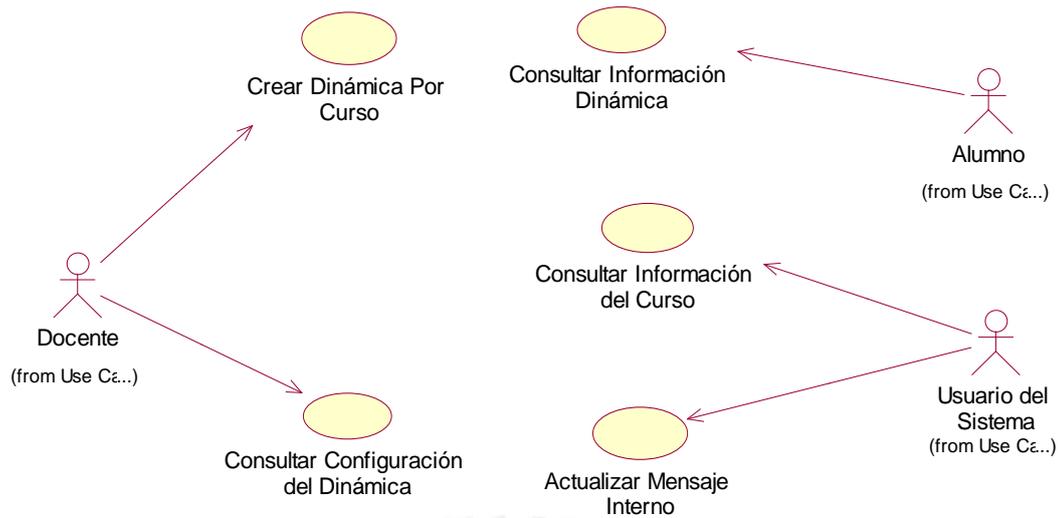


Figura 8: Diagrama de Casos de Usos del Paquete Planificación

En la Figura 8 se puede apreciar que el Docente tendrá a su cargo la creación de la Dinámica (Caso de Uso: “Crear Dinámica Por Curso”) y le permitirá consultar la configuración de la Dinámica creada (Caso de Uso: “Consultar Configuración de la Dinámica”). Puesto que este paquete pertenece al módulo de Planificación, el Alumno solo podrá realizar la consulta de la información de la dinámica creada (Caso de Uso: “Consultar Información Dinámica”).

En la Figura 8 también se puede observar que el Usuario del Sistema podrá acceder a la información de los cursos que está asignado (Caso de Uso: “Consultar Información del Curso”), y también podrá hacer uso del servicio de mensajería del Sistema (Caso de Uso: “Actualizar Mensaje Interno”).

A continuación en el Cuadro 8 se presenta la descripción de las funcionalidades de cada Caso de Uso que forma parte del Paquete de Planeamiento.

CASO DE USO	DESCRIPCIÓN
Crear Dinámica por Curso	Este Caso de Uso permite al Docente registrar y configurar en el Sistema una nueva dinámica la cual estará relacionada a un curso.
Consultar Información del Curso	Este caso de uso permite al Usuario consultar la información de un curso.
Consultar Configuración de Dinámica	Este caso de uso permite al Docente consultar la información de una Dinámica que ha creado.
Consultar Información de Dinámica	Este caso de uso permite al Alumno consultar la información de una Dinámica asociada a un curso que está inscrito.
Actualizar Mensaje Interno	Este caso de uso permite al Usuario realizar el mantenimiento de los mensajes internos dentro de su sesión. Esto incluye revisar, responder, crear y eliminar un mensaje.

Cuadro 8: Casos de Uso Paquete Planificación

En el Cuadro 8 se puede apreciar una descripción más detallada de las funcionalidades de los Casos de Uso descritos en la Figura 8. De este cuadro, se puede identificar como el Caso de Uso más representativo a “Crear Dinámica por Curso”; ya que en este se realiza la creación y configuración de la información y las etapas que realizarán los Alumnos asignados a una sesión cooperativa.

Paquete Ejecución

Este paquete se encarga de llevar a cabo la ejecución de una sesión basada en la Técnica de Aprendizaje Cooperativo de Rompecabezas. Permite el desarrollo paso a paso desde la lectura de materiales y documentos, hasta la elaboración cooperativa de la solución en los diferentes Grupos *Jigsaw* y *Expertos*.

Se contará con herramientas de comunicación en línea como Chat, las cuales permitirán la interacción para la coordinación de la edición cooperativa; además se contará con Foros de Discusión para complementar el desarrollo y esclarecimiento de algunas dudas sobre estos temas propuestos.

A continuación en la Figura 9 se presenta el Diagrama de Casos de Uso del Módulo de Ejecución.

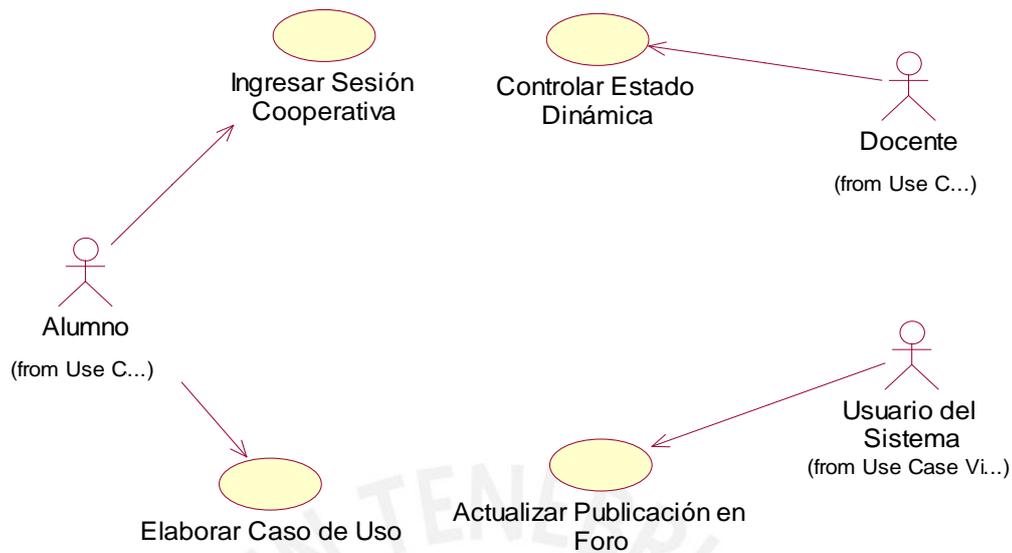


Figura 9: Diagrama de Casos de Usos del Módulo de Ejecución

En la Figura 9 se puede apreciar que el Docente tendrá a su cargo el control de los estados de la Dinámica (Caso de Uso: “Controlar Estado Dinámica”), es decir, el Docente decidirá en que momento inicia o concluye una etapa de la Dinámica (basándose en la configuración registrada para esa Dinámica). El Alumno por su parte podrá tener acceso a la Sesión Cooperativa de la Dinámica (Caso de Uso: “Ingresar Sesión Cooperativa”) y elaborar Casos de Uso con los integrantes de la Sesión (Caso de Uso: “Elaborar Caso de Uso”).

En la Figura 9 también se puede observar que el Usuario del Sistema podrá emplear el Foro de Discusión de la Dinámica en la que participa (Caso de Uso: “Actualizar Publicación en Foro”).

A continuación en el Cuadro 9, se presenta la descripción de las funcionalidades de cada Caso de Uso que forman parte del Paquete de Ejecución.

CASO DE USO	DESCRIPCIÓN
Actualizar Publicación en Foro	Este caso de uso permite al Usuario realizar el mantenimiento de las publicaciones dentro del Foro de Discusión. Esto incluye revisar, responder, crear y eliminar una pregunta.
Ingresar Sesión Cooperativa	Este Caso de uso permite al Alumno ingresar a la Sesión Cooperativa de una Dinámica asignada.
Controlar Estado Dinámica	Este Caso de Uso permite al Docente iniciar/finalizar un estado que puede tomar una Dinámica.
Elaborar Caso de Uso	Este Caso de uso permite al Alumno gestionar cooperativamente el contenido de un Caso de Uso durante el proceso de elaboración del Caso Propuesto en la Dinámica.

Cuadro 9: Casos de Uso Paquete Ejecución

En la Cuadro 9 se puede apreciar una descripción más detallada de las funcionalidades de los Casos de Uso descritos en la Figura 9. De este cuadro, se puede identificar como el Caso de Uso más representativo a “Elaborar Caso de Uso”; ya que en este se emplea el Espacio Cooperativo para la elaboración de Casos de Uso. Aquí se detalla las interacciones con esta herramienta y los integrantes de la Sesión.

Paquete Evaluación

Este paquete se encarga de la elaboración de preguntas y evaluaciones por parte del tutor y docente; y posteriormente de la ejecución de la evaluación por parte de los alumnos para complementar el desarrollo de una sesión. Así mismo brinda la posibilidad de calificación manual y automática de las evaluaciones rendidas.

A continuación en la Figura 10 se presenta el Diagrama de Casos de Uso del Módulo de Evaluación.

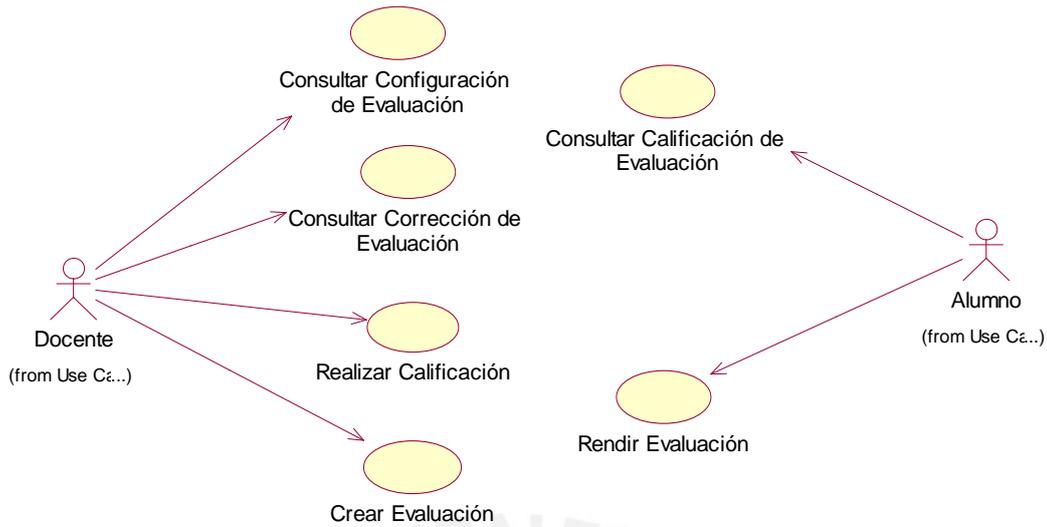


Figura 10: Diagrama de Casos de Usos del Módulo de Evaluación

En la Figura 10 se puede apreciar que el Docente podrá crear una Evaluación asignada a una Dinámica, mediante el ingreso de enunciados, preguntas y sus respectivas respuestas (Caso de Uso: “Crear Evaluación”); también podrá revisar la configuración de las preguntas y respuestas de la Evaluación creada (Caso de Uso: “Consultar Configuración de Evaluación”). Finalmente, el Docente podrá realizar la calificación de una evaluación concluida por el Alumno (Caso de Uso: “Realizar Calificación”) y también realizar una revisión de una calificación realizada a una Evaluación (Caso de Uso: “Consultar Corrección Evaluación”).

En la Figura 10 también se puede observar que el Usuario rinde la Evaluación respondiendo las preguntas asignadas a esta (Caso de Uso: “Rendir Evaluación”) y las registra en el Sistema. Posteriormente, el alumno puede acceder al detalle de la calificación y corrección de la Evaluación que ha rendido (Caso de Uso: “Consultar Calificación de Evaluación”).

A continuación en el Cuadro 10, se presenta la descripción de las funcionalidades de cada Caso de Uso que forma parte del Paquete de Evaluación.

CASO DE USO	DESCRIPCIÓN
Consultar Configuración de Evaluación	Este Caso de Uso permite al Docente revisar la configuración de una Evaluación creada previamente.
Consultar Corrección de Evaluación	Este Caso de Uso permite al Docente revisar la corrección de una Evaluación de un Alumno que forma parte de la Dinámica.
Consultar Calificación de Evaluación	Este Caso de Uso permite al Alumno consultar su calificación de una Evaluación.
Crear Evaluación	Este Caso de Uso permite al Docente elaborar y configurar una Evaluación asociada a una Dinámica.
Rendir Evaluación	Este Caso de Uso permite al Alumno rendir una evaluación asociada a una Dinámica.
Realizar Calificación	Este Caso de Uso permite al Docente realizar la calificación de las Evaluaciones de una Dinámica.

Cuadro 10: Casos de Uso Paquete Evaluación

En la Cuadro 10 se puede apreciar una descripción más detallada de las funcionalidades de los Casos de Uso descritos en la Figura 10. De este cuadro, se puede identificar como el Caso de Uso más representativo a “Rendir Evaluación”; ya que en este el Alumno realiza la evaluación asignada a la Dinámica en la que participa.

2.3.4. Requisitos Específicos: Especificación de Casos de Uso

Esta sección contiene a nivel detallado la funcionalidad y restricciones del sistema de las principales funcionalidades de cada uno de los módulos del sistema. El detalle de la funcionalidad se determina con la especificación de casos de uso.

A continuación se detallarán un Caso de Uso por cada Módulo que forma parte del Sistema Web *Jigsaw* (Módulo de Planificación, Módulo de Ejecución y Módulo de Evaluación). Para mayor detalle sobre los Caso de Uso del Sistema revisar el “Anexo I – Especificación de Requisitos de Software”.

A. Módulo de Planificación

A continuación se describirá el detalle del Caso de Uso “Crear Dinámica por Curso”, el cual permite al Docente registrar y configurar en el Sistema una nueva dinámica la cual estará relacionada a un curso.

Nombre	Crear Dinámica por Curso
Actor	Docente
Breve Descripción	Este Caso de Uso permite al Docente registrar y configurar en el Sistema una nueva dinámica la cual estará relacionada a un curso.
Precondiciones	El usuario ha sido admitido en el Sistema
Flujo básico	<ol style="list-style-type: none"> 1. El Caso de Uso se inicia cuando el Docente accede a la página principal de su sesión de usuario. 2. El Sistema presenta una lista de los cursos asignados al Docente que inicia la sesión. La información que muestra cada curso que conforma esta lista será: Nombre Curso, Horario del Curso, Profesor asignado al Horario y la opción “Crear Dinámica” para dicho curso. 3. El Docente selecciona la opción “Crear Dinámica” del curso que desea crear una nueva dinámica. 4. El Sistema muestra una lista con los temas que conforman el curso. 5. El Docente elige el tema principal de será la base de esta dinámica. 6. El Sistema muestra un formulario principal vacío con los siguientes datos informativos: Nombre del Curso, Horario del Curso y Tema del Curso. Además, este formulario contendrá cajas de texto vacías para ingresar el Nombre de la Sesión, Descripción de la Sesión, Fecha de la Sesión, Hora de la Sesión, Duración de la Dinámica y Duración de la Evaluación. 7. El Docente ingresa los datos requeridos por el

- formulario.
8. El Sistema muestra una lista con los Temas Expertos registrados junto a la opción "Crear Tema Experto". La información que se muestra por cada Tema Experto será: Nombre del Tema Experto y la opción "Borrar el Tema Experto".
 9. El Docente selecciona la opción "Crear Tema Experto".
 10. El Sistema muestra un formulario que presenta cajas de texto vacías para ingresar el Nombre y la Descripción del Tema a experto a registrar, además de la opción "Guardar Datos".
 11. El Docente llena los datos requeridos y selecciona la opción "Guardar Datos".
 12. El Sistema actualiza el contenido de la Lista de Temas Expertos incluyendo los últimos datos registrados.
 13. El Sistema muestra una lista con los Documentos registrados junto a la opción "Crear Documento". La información a mostrar por cada Documento será: Nombre del Documento y la opción "Borrar Documento".
 14. El Docente selecciona la opción "Crear Documento".
 15. El Sistema muestra un formulario que presenta cajas de texto vacías para ingresar el Nombre y la Descripción del Documento a registrar, un buscador de archivos, además de la opción "Guardar Datos".
 16. El Docente llena los datos requeridos y selecciona la opción "Guardar Datos".
 17. El Sistema actualiza el contenido de la Lista de Documentos incluyendo los últimos datos registrados.
 18. El Docente selecciona del formulario principal la opción "Guardar Configuración".

	<ol style="list-style-type: none"> 19. El Sistema presenta espacios rectangulares vacíos con identificadores referentes a cada grupo creado. 20. El Sistema presenta la lista de Alumnos matriculados en el curso identificados con sus nombres/apellidos. 21. El Docente procede a arrastrar la identificación de cada Alumno a los espacios rectangulares que identifican a cada grupo. 22. El Sistema internamente realiza la asociación de los Alumnos a cada grupo donde es arrastrado. 23. Se repite los pasos 21-22 hasta que el Docente haya realizado la asignación de todos los Alumnos de la lista. Cuando se termina la asignación, el Docente selecciona la opción “Guardar Datos”. 24. El Sistema agrega la asignación de Alumnos a la configuración guardada de la dinámica en creación.
Flujo alterno	<p>Ver Detalle Tema Experto</p> <ol style="list-style-type: none"> 1. El Docente selecciona uno de los Temas Expertos de la lista que presenta el formulario principal de Creación de Dinámica. 2. El Sistema muestra en pantalla el Título del Tema Experto, la Descripción del mismo y la opción “Aceptar”. 3. El Docente revisa la información presentada y luego selecciona la opción “Aceptar”. 4. El Sistema cierra la pantalla de informativa del Tema Experto seleccionado. <p>Eliminar Tema Experto</p> <ol style="list-style-type: none"> 1. El Docente selecciona la opción “Borrar Tema Experto” de uno de los Temas Expertos de la lista que presenta el formulario principal de

Creación de Dinámica.

2. El Sistema solicita la confirmación de los cambios realizados.
3. El Docente confirma las modificaciones.
4. El Sistema guarda los cambios realizados y se actualiza el contenido en la lista de Temas Expertos.
5. Los pasos 1-4 son repetidos para cada Tema Experto que se desee eliminar. Cuando se termina de eliminar los Temas Expertos que el Docente desea, el caso de uso finaliza.

Ver Detalle Documento

1. El Docente selecciona uno de los Documentos de la lista que presenta el formulario principal de Creación de Dinámica.
2. El Sistema muestra en pantalla el Título del Documento, la Descripción del mismo, así como el nombre del Archivo correspondiente.
3. El Docente revisa la información presentada y luego selecciona la opción "Aceptar".
4. El Sistema cierra la pantalla de informativa del Documento seleccionado.

Eliminar Documento

1. El Docente selecciona la opción "Borrar Documento" de uno de los Documentos de la lista que presenta el formulario principal de Creación de Dinámica.
2. El Sistema solicita la confirmación de los cambios realizados.
3. El Docente confirma las modificaciones.
4. El Sistema guarda los cambios realizados y se actualiza el contenido en la lista de Documentos.
5. Los pasos 1-4 son repetidos para cada Documento que se desee eliminar. Cuando se

	termina de eliminar los Documentos que el Docente desea, el caso de uso finaliza.
Flujo Excepcional	<p>Cancelar</p> <ol style="list-style-type: none"> 1. En cualquier parte del flujo principal o del flujo alterno el Docente puede seleccionar “Cancelar”. 2. El Sistema muestra un mensaje de verificación. 3. Si el usuario selecciona “Sí” el caso de uso termina. 4. Si el Docente selecciona “No” el caso de uso continúa donde se quedó. <p>Datos incorrectos</p> <ol style="list-style-type: none"> 1. Si el Sistema detecta que los datos ingresados son incorrectos muestra un mensaje de error. 2. Una vez que el usuario los corrija el caso de uso continúa y se podrá guardar la información. <p>Datos obligatorios no registrados</p> <ol style="list-style-type: none"> 1. Si el Sistema detecta que no se han llenado todos los campos obligatorios del formulario muestra el mensaje "Falta completar los campos obligatorios". 2. Una vez que el Docente llene los campos que le faltaban el caso de uso continúa y se podrá guardar la información.
Poscondición	El Docente registra satisfactoriamente la dinámica de un curso
Puntos de extensión	No presenta Puntos de Extensión

B. Módulo de Ejecución

A continuación se describirá el detalle del Caso de Uso “Elaborar Caso de Uso”, el cual permite al Alumno gestionar cooperativamente el contenido de un Caso de Uso durante el proceso de elaboración del Caso Propuesto en la Dinámica.

Nombre	Elaborar Caso de Uso
Actor	Alumno
Breve Descripción	Este Caso de uso permite al Alumno gestionar cooperativamente el contenido de un Caso de Uso durante el proceso de elaboración del Caso Propuesto en la Dinámica.
Precondiciones	El Alumno ha sido admitido en el Sistema.
Flujo básico	<ol style="list-style-type: none"> 1. El Caso de Uso se inicia cuando el Alumno selecciona la opción para ingresar a la sesión cooperativa de una Dinámica asignada. 2. El Sistema presenta la Herramienta Cooperativa para elaboración de los Casos de Uso. Esta está conformada por un editor de texto definido como “Área de Trabajo” (con las opciones de un procesador de texto: Tipo de Letra, Tamaño de Letra, Color, Formatos de Letra); por una lista que muestra a los Usuarios que participan de la Sesión; y un chat que permitirá la comunicación entre estos. 3. El Alumno ingresa un texto en el “Área de Trabajo”. 4. El Sistema actualiza los contenidos ingresados en todas las estaciones participantes según reglas de concurrencia; y el Caso de Uso finaliza.
Flujo alternativo	<p>Copiar y Pegar Texto</p> <ol style="list-style-type: none"> 1. El Alumno selecciona un texto de la herramienta de Espacio Cooperativo para elaboración de los Casos de Uso o de alguna

fuerza externa al Sistema.

2. El Sistema almacena temporalmente en la sesión del Usuario el contenido seleccionado.
3. El Alumno realiza el pegado de esta información en el área de trabajo de la herramienta de Espacio Cooperativo para elaboración de los Casos de Uso.
4. El Sistema actualiza los contenidos ingresados en todas las estaciones participantes según reglas de concurrencia, y el Caso de Uso finaliza.
5. Los pasos 1-4 son repetidos para cada contenido que el Usuario desee copiar y pegar. Cuando se termina de copiar y pegar los contenidos que el Usuario desea, el caso de uso finaliza.

Eliminar Texto

1. El Alumno selecciona un texto de la herramienta de Espacio Cooperativo para elaboración de los Casos de Uso.
2. El Sistema identifica el contenido del texto seleccionado en la sesión del Usuario.
3. El Alumno elige la opción del borrado del contenido seleccionado del área de trabajo de la herramienta de Espacio Cooperativo para elaboración de los Casos de Uso.
4. El Sistema actualiza los contenidos ingresados en todas las estaciones participantes según reglas de concurrencia.
5. Los pasos 1-4 son repetidos para cada contenido que el Usuario desee copiar y pegar. Cuando se termina de copiar y pegar los contenidos que el Usuario desea, el caso de uso finaliza.

	<p>Almacenar Texto en archivo físico</p> <ol style="list-style-type: none"> 1. El Alumno realiza el ingreso de texto en el Área de Trabajo de la Herramienta de Espacio Cooperativo para elaboración de los Casos de Uso, y selecciona la opción “Guardar Contenido”. 2. El Sistema presenta una caja de texto donde se solicita el nombre del archivo a crear. 3. El Alumno ingresa el dato solicitado. 4. El Sistema guarda los contenidos del Área de Trabajo en un archivo físico y permite la descarga del mismo 5. Los pasos 1-4 son repetidos para cada contenido que el Usuario desee almacenar en un archivo físico. Cuando se termina de almacenar los contenidos que el Usuario desea, el caso de uso finaliza.
<p>Flujo Excepcional</p>	<p>Cancelar</p> <ol style="list-style-type: none"> 1. En cualquier parte del flujo principal o del flujo alternativo el usuario puede seleccionar “Cancelar”. 2. El Sistema muestra un mensaje de verificación. 3. Si el usuario selecciona “Sí” el caso de uso termina. 4. Si el usuario selecciona “No” el caso de uso continúa donde se quedó.
<p>Poscondición</p>	<p>La actualización del contenido del Área de Trabajo se actualiza satisfactoriamente en todas las estaciones participantes.</p>
<p>Puntos de extensión</p>	<p>No presenta Flujos de Extensión.</p>

C. Módulo de Evaluación

A continuación se describirá el detalle del Caso de Uso “Rendir Evaluación”, el cual permite al Alumno rendir una evaluación asociada a una Dinámica.

Nombre	Rendir Evaluación
Actor	Alumno
Breve Descripción	Este Caso de Uso permite al Alumno rendir una evaluación asociada a una Dinámica.
Precondiciones	El Alumno debe haber finalizado una Sesión Cooperativa.
Flujo básico	<ol style="list-style-type: none"> 1. El Caso de Uso se inicia cuando el Usuario accede a la página principal de su sesión de usuario. 2. El Sistema presenta una lista con las Dinámicas asignadas al Usuario que inicia la sesión. La información que muestra cada Dinámica que conforma esta lista será: Nombre de la Dinámica, Nombre del Curso, Fecha y Hora en que se realizará la Dinámica, Duración de la Dinámica, el Estado actual en que se encuentra la Dinámica (“Pendiente”, “Grupo Experto”, “Grupo Jigsaw”, “Rendir Evaluación”) y la opción “Ir A Foro”. 3. El Alumno selecciona una de las dinámicas de la lista. 4. El Sistema muestra los siguientes datos informativos: Nombre del Curso, Horario del Curso, Tema del Curso, Nombre de la Dinámica, Descripción de la Dinámica, Título y Descripción del Tema Experto Asignado. Adicionalmente se muestra un menú indicando los estados que puede tomar la dinámica seleccionada (“Grupo Experto”, “Grupo Jigsaw” y “Rendir Evaluación”), y un listado mostrando la documentación asignada a la Dinámica Elegida. La información que se

- muestra en este listado es: Título del Documento, Descripción del Documento y el Nombre del Archivo referente al Documento.
5. El Alumno seleccionará del Menú de Estados de la Dinámica el estado "Rendir Evaluación".
 6. El Sistema muestra en pantalla lo siguiente:
 - Una lista con las preguntas que forman parte la evaluación. Cada pregunta presenta un enlace para indicar que se debe mostrar en pantalla sus contenidos y respuestas.
 - Los detalles de la Evaluación seleccionada: el Nombre del Curso, Horario del Curso, Nombre del Tema del Curso, Nombre de la Dinámica, Enunciado Inicial de la Evaluación,
 - El enunciado de la pregunta seleccionada y un listado de respuestas disponibles. Para el caso que la pregunta sea de Tipo "Desarrollada", solo se muestra un enlace para adjuntar un archivo conteniendo la respuesta a esa pregunta.
 7. El Alumno selecciona una de las preguntas de la lista presentada por el Sistema.
 8. El Sistema muestra en pantalla los contenidos referentes a la pregunta seleccionada.
 9. El Alumno selecciona una respuesta (en caso la pregunta sea de "Opción Simple"), más de una respuesta (en caso la pregunta sea de "Opción Múltiple"), o adjunta un archivo de respuesta (en caso la pregunta sea "Desarrollada"), para la pregunta que se ha seleccionado.
 10. El Sistema almacena en sesión las respuestas ingresadas por el Alumno.
 11. Los pasos 7-10 son repetidos para cada

	<p>pregunta que el Alumno desee responder. Cuando se termina de responder las preguntas que el Alumno desea, este selecciona la opción “Registrar Respuestas”.</p> <p>12. El Sistema registra las respuestas registradas por el Alumno, y el Caso de Uso finaliza.</p>
Flujo alterno	No presenta Flujo Alterno
Flujo Excepcional	No presenta Flujo Excepcional
Poscondición	El Alumno rinde satisfactoriamente una Evaluación.
Puntos de extensión	No presenta Flujos de Extensión.

2.3.5. Requisitos Específicos: Diagrama de Clases de Análisis

En esta sección se detallará el Diagrama de Clases del Sistema Web *Jigsaw*. Un Diagrama de Clases de Análisis es un tipo de diagrama estático UML que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Los Diagramas de Clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

A continuación en la Figura 11, se presenta el detalle del Diagrama de Clases para el Sistema Web *Jigsaw*.

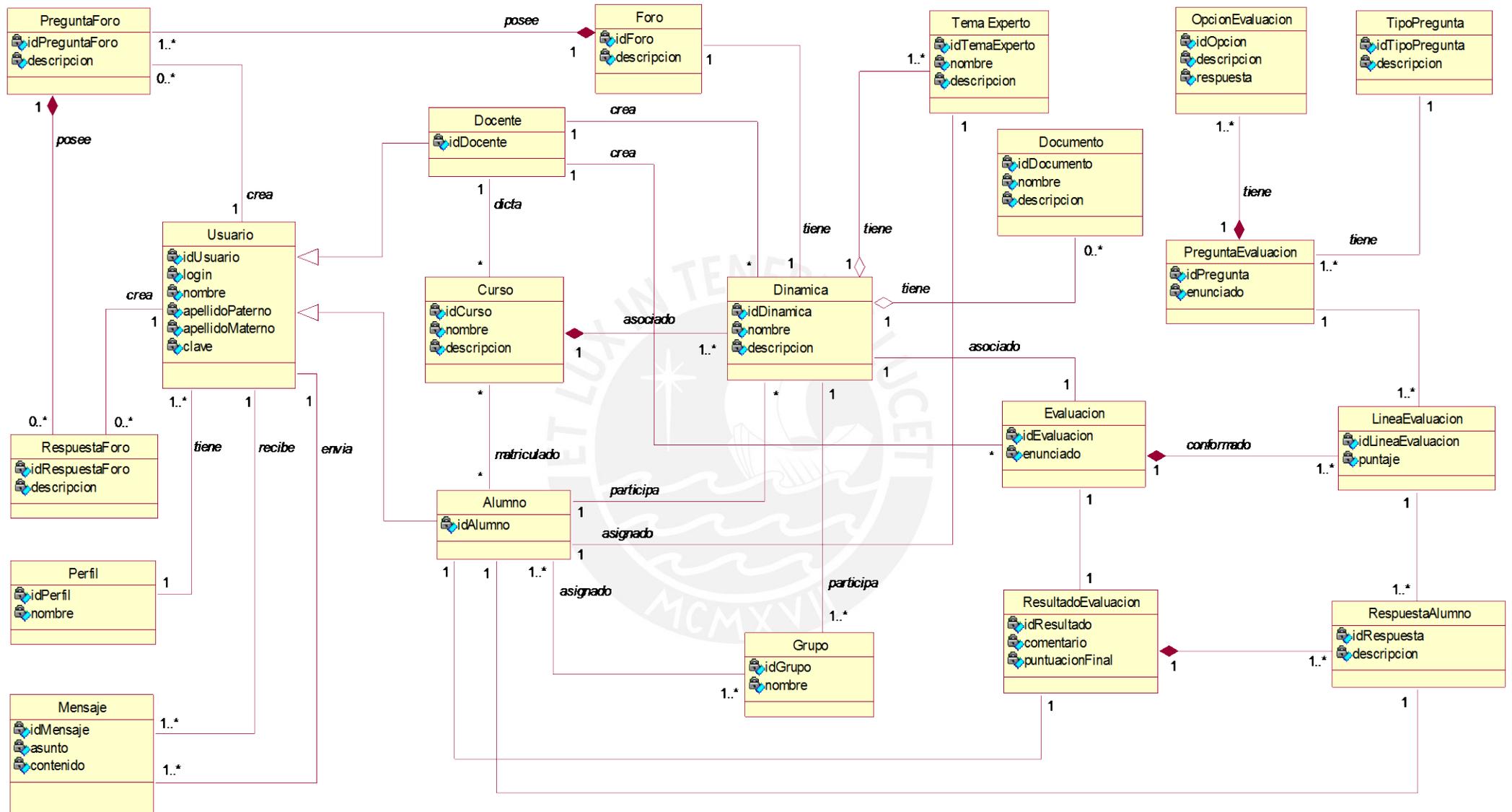


Figura 11: Diagrama de Clases del Sistema Web Jigsaw

Tal como se puede observar en el Diagrama de Clases de la Figura 11:

El Sistema Web *Jigsaw* permite el acceso a Usuarios registrados en su Sistema (relación con la Clase "Usuario"), a los cuales se les asigna un perfil de acceso (relación con la Clase "Perfil"). Dentro del Sistema el usuario puede enviar mensajes internos a otros usuarios (relación con la Clase "Mensaje") y participar de los Foros de Discusión (relación con la Clase "Foro") que esté asignado.

Como Usuario del Sistema se identifican dos roles: el de Docente (Clase "Docente", que hereda de la Clase "Usuario") y el de Alumno (Clase "Alumno", que hereda de la Clase "Usuario").

El Docente (relación con la Clase "Docente") tiene asignados Cursos para dictar (relación con la Clase "Curso"). En cada uno de estos cursos asignados puede crear Dinámicas (relación con la Clase "Dinámica"), las cuales tendrán asignadas una evaluación respectivamente (relación con la Clase "Evaluación").

El alumno se matricula en un curso (relación con la Clase "Curso") y por ello participa de una Dinámica (relación con la Clase "Dinámica"). Al participar en una Dinámica forma parte de un grupo (relación con la Clase "Grupo") y se le asigna una tarea previa a la Dinámica llamada "Tema Experto" (relación con la Clase "Tema Experto") y se le pone a su disposición material de lectura previo a la Dinámica (relación con la Clase "Documento"). Al concluir la Dinámica, el alumno participa de una Evaluación (relación con la Clase "Evaluación"), para consolidar la información brindada en la Dinámica.

Una Dinámica (relación con la Clase "Dinámica") tiene asignado un Foro de Discusión (relación con la Clase "Foro") referente al tema de la Dinámica. En estos, el Usuario del Sistema realiza una pregunta (relación con la Clase "Pregunta Foro"), la cual obtendrá respuestas (relación con la Clase "Respuesta Foro") de otros usuarios asignados a ese Foro de Discusión.

Finalmente una Dinámica tiene asignada una Evaluación (relación con la Clase "Evaluación"), la cual consta de varias preguntas (relación con la Clase "Pregunta Evaluación") las cuales pueden ser "Opción Simple", "Opción Múltiple" o "Pregunta Desarrollada" (relación con la Clase "Tipo Pregunta"). Esta pregunta presenta una lista de opciones para elegir como respuesta correcta (relación con

la Clase Opción “Evaluación”). Cada respuesta respondida por el alumno (relación con la Clase “Respuesta Alumno”) referente a una determinada pregunta (relación con la Clase “Línea Evaluación”) es finalmente almacenada en el Sistema para su posterior calificación por el Docente (relación con la Clase “Resultado Evaluación”).



Capítulo III: Diseño

En el presente capítulo se describirá la etapa de diseño del proyecto las cuales se enfocan a la arquitectura de la solución y al diseño de interfaz gráfica.

La primera sección a tratar es la arquitectura de la solución la cual muestra las diferentes perspectivas, metas y restricciones a nivel técnico a los que se estará expuesto en el proyecto. A continuación, se describe las vistas de la arquitectura a utilizar en el desarrollo del sistema como base en las disciplinas que define RUP.

Posteriormente, como segunda y última sección de este capítulo se hará un recorrido a través de las principales pantallas del sistema y se describirán su diseño y distribución.

3.1. Arquitectura de la Solución

Esta sección provee una vista general de alto nivel del desarrollo de la arquitectura técnica para el Sistema Web *Jigsaw*. La necesidad de definir una arquitectura se fundamenta en lo siguiente:

- El programador requiere de una estructura para el desarrollo del sistema que facilite su trabajo identificando las partes que lo conforman.
- El sistema podría requerir mantenimientos, ampliaciones o modificaciones en el futuro por lo que se debe proveer una base que facilite dicha labor.

El análisis y diseño será plasmado a través de un conjunto de vistas que permitirán mostrar tanto el aspecto funcional y el aspecto técnico propuesto para el Sistema Web *Jigsaw*. Las vistas de la arquitectura que se detallarán en este capítulo son la vista lógica, la vista de implementación y la vista de despliegue

3.1.1. Metas Arquitectónicas y Restricciones

La meta del Sistema Web *Jigsaw* es que opere sobre una plataforma Web. Para que esto sea posible es necesario utilizar las herramientas y tecnologías que permitan crear una arquitectura Web ventajosa a pesar de las limitaciones de equipos que se tenga.

El proyecto solo cuenta con un servidor de aplicaciones *JBoss* y *Servidor Web Apache HTTP*. La base de datos es *MySQL*. Ambos equipos estarán disponibles las 24 horas del día. El mantenimiento físico de estos componentes se encuentra fuera del proyecto, por lo que no se tendrá en cuenta ese aspecto.

Plataforma Técnica

La base técnica para el desarrollo del sistema se encuentra ya concebida y en funcionamiento desde antes del proyecto. Se cuenta con un servidor de aplicaciones *JBoss* y *Servidor Web Apache HTTP* para tecnologías *javas* y el software para la base de datos *MySQL* con las siguientes características:

- Completa integración con el Sistema Operativo Linux permitiendo procesamiento multihilo y multiprocesamiento simétrico.
- Protección completa de integridad de los datos, sistema de seguridad a nivel de aplicación, bases de datos y objetos.
- Altas prestaciones a bajos costos en relación con otros productos similares.
- Soporte para bases de datos de gran volumen.

Transacción

Las transacciones serán manejadas por la arquitectura que se implementa al emplear la librería para Mapeo de Datos *MyBatis*. Esta se encargará de mantener la integridad de los datos, haciendo que las transacciones no puedan finalizar en un estado intermedio. También manejará la integridad en caso de fallas en la base de datos, o por si sucede algún otro altercado por el cual el sistema deba de cancelar las transacciones. Para esto deberá deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial (llamado punto de integridad), como si la orden de las transacciones nunca se hubiese realizado.

Persistencia

La persistencia se logrará utilizando una base de datos relacional. La arquitectura implementada para el uso de la librería para Mapeo de Datos *MyBatis* ayudará con la persistencia de los objetos en la ejecución de la aplicación.

Seguridad

La seguridad del sistema estará basada en perfiles. La aplicación contendrá los siguientes puntos:

- Autenticación: Cada usuario deberá identificarse previamente en el sistema para poder ingresar, de lo contrario no podrá tener acceso a ninguna funcionalidad.
- Autorización: Cada perfil tendrá distintos permisos, que le permitirán ingresar a diferentes partes del sistema.

3.1.2. Arquitectura

A continuación se describe las vistas de la arquitectura a utilizar en el desarrollo del sistema como base en las disciplinas que define RUP.

A. Vista de Casos de Uso

La Vista de Casos de Uso es la encargada de presentar la percepción que tiene el usuario de las funcionalidades del sistema; presentando a los actores y los casos de uso que participan.

Para detallar la presente Vista se mostrará un solo Caso de Uso por cada Módulo del Sistema Web *Jigsaw*. Para mayor detalle sobre los Caso de Uso del Sistema revisar el Anexo I – Especificación de Requisitos de Software.

- Crear Dinámica por Curso (Módulo de Planificación).
- Ingresar a Sesión Cooperativa (Módulo de Ejecución).
- Crear Evaluación (Módulo de Evaluación).

Estos casos de uso son inicializados por el Docente y alumno. A continuación en la figura 12, se presenta el Diagrama de Casos de Uso para esta Vista.

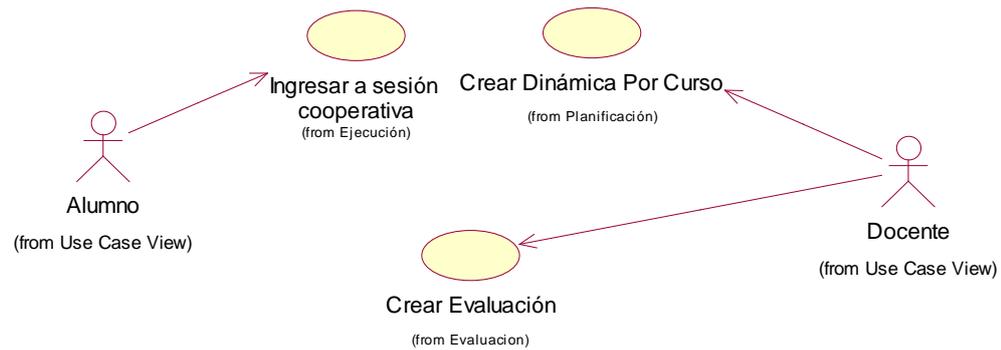


Figura 12: Vista de Casos de Uso

- **Crear Dinámica por curso**
Este caso de uso permitirá al docente la creación de las dinámicas *Jigsaw* de un curso.
- **Ingresar a Sesión Cooperativa**
Este Caso de uso se encarga de mostrar la etapa cooperativa actual que se está desarrollando y permite el acceso al Espacio Cooperativo para elaboración de los Casos de Uso
- **Crear Evaluación**
Este caso de uso permitirá al docente la elaboración y configuración de la evaluación asociada a una dinámica.

B. Vista Lógica

La Vista Lógica es la encargada de presentar lo que el sistema debe de hacer, las funciones y servicios que se han definido.

Para detallar la presente Vista, la solución presentada estará dividida en capas. El modelo se basa en una estrategia que asocia una determinada responsabilidad del funcionamiento del Sistema Web *Jigsaw* a cada capa. Se ha elegido esta estrategia pues se sigue el Modelo Vista Controlador, el cual permite aislar y modular las responsabilidades; y en consecuencia su desarrollo.

A continuación en la figura 13, se presenta el Diagrama que representa a esta Vista.

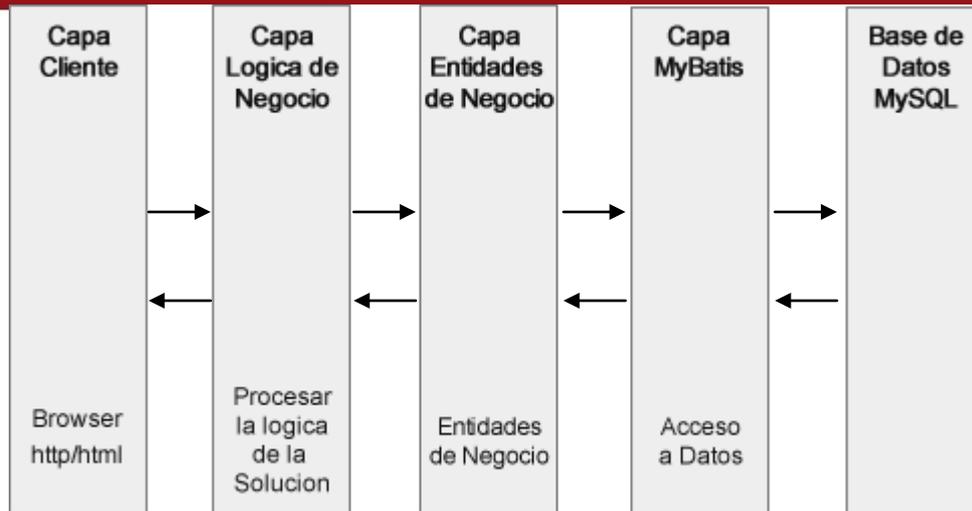


Figura 13: Vista Lógica

- **Capa Cliente**

La capa Cliente aloja el browser que se encargará de mostrar las páginas web que genera el servidor de aplicaciones. Esta capa realiza las peticiones y/o solicitudes requeridas por el usuario y muestra el resultado de las mismas obtenido de la capa de lógica del negocio.
- **Capa Lógica del Negocio**

La lógica del negocio está representada por las clases que se encargan de manejar la información mediante las reglas de negocio establecidas por el sistema, es por ello que presenta una dependencia hacia la capa de entidades de negocio. Esta capa representa el límite entre el cliente y la capa media.
- **Capa de Entidades del Negocio**

La capa de Entidades del Negocio contiene a todas las clases que representan las entidades del Sistema. Estas residen en el servidor y son usadas por la capa de Lógica del Negocio.
- **Capa MyBatis**

La capa *MyBatis* convertirá y administrará los datos utilizados por Java y los definidos por SQL. Esta capa contiene lo necesario para la interconexión hacia el servidor de bases de datos.
- **Capa de Base de Datos MySQL**

La capa de Base de Datos contiene los objetos persistentes del Sistema.

C. Vista de Implementación

La Vista de implementación es la encargada de describir la asignación de paquetes y clases de la Vista Lógica, a los paquetes y módulos de la Vista de Implementación.

- **jigsaw.view:**
Contiene todos los componentes que permitirán al usuario interactuar con el sistema. Está conformado por el navegador web y las páginas del Sistema. Este paquete es el que actúa directamente con el *Framework Struts 2*.
- **STRUTS 2:**
Contiene todos los componentes que permiten al *Framework Struts 2* interpretar los pedidos realizados por el paquete “jigsaw.view” y re direccionar a una acción mapeada a ese pedido (paquete: “jigsaw.actions”).
- **jigsaw.actions:**
Contiene las clases que implementan la lógica de negocio del sistema. Estas construyen su lógica de negocio mediante invocaciones a clases que manejan funcionalidades definidas para las entidades que conforman el sistema (paquete: “jigsaw.controller”).
- **jigsaw.controller:**
Contiene las clases que implementan la funcionalidad de una entidad dentro del Sistema. Estas funcionalidades también pueden incluir invocaciones a la Base de Datos por medio de clases del paquete “jigsaw.model”.
- **jigsaw.model:**
Contiene las clases que implementan funcionalidades y devuelven información de la base de datos por medio de la librería MyBatis. Estas clases emplean clases auxiliares de mapeo de información (paquete: “jigsaw.model.mybatis”).
- **MyBatis:**
Contiene todos los componentes que permiten a la librería MyBatis poder comunicarse con la Base de Datos MySQL y traer información a las clases del paquete “jigsaw.model”.

A continuación en la figura 14, se presenta el Diagrama que representa a esta Vista.

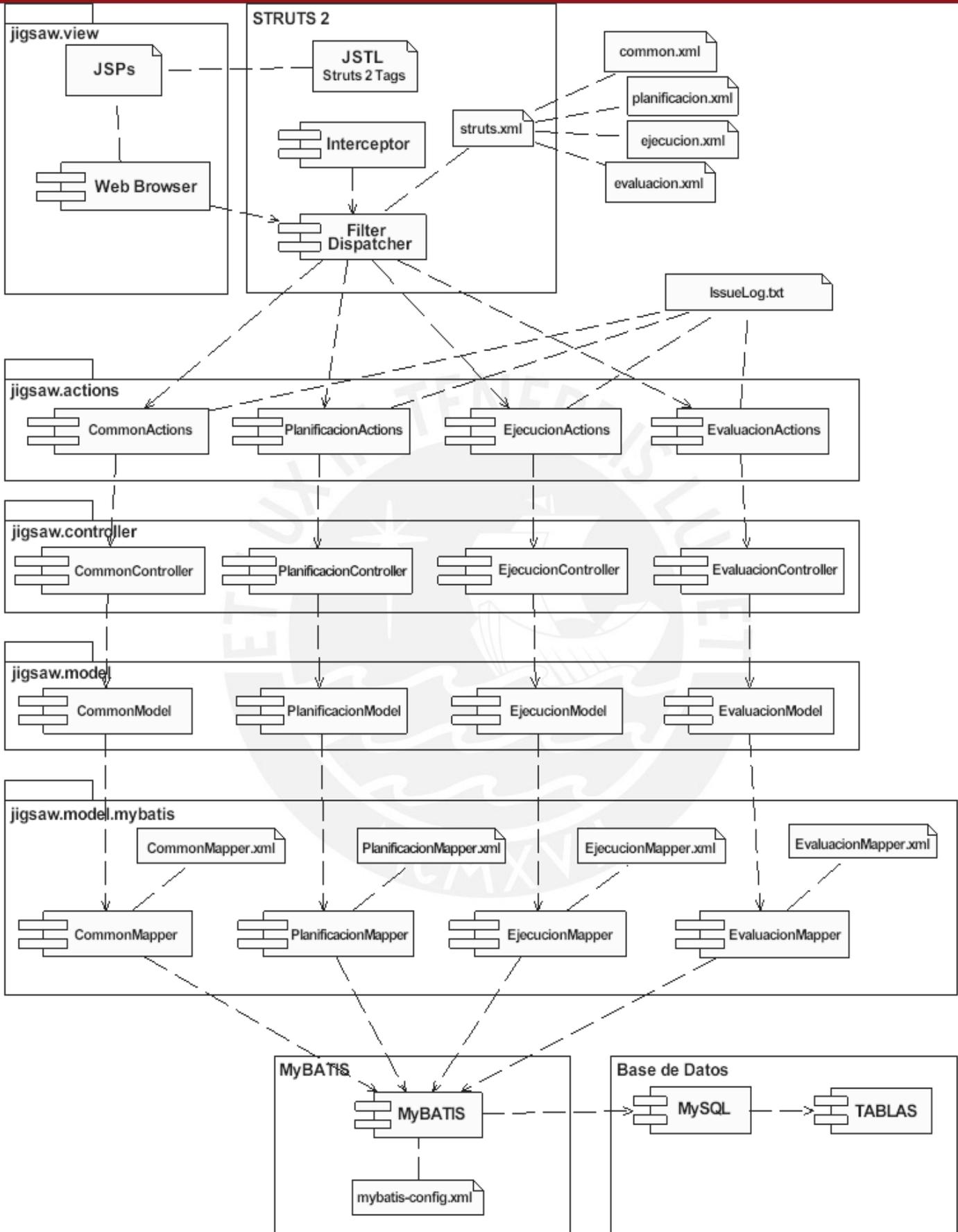


Figura 14: Diagrama de Componentes

D. Vista de Despliegue

Esta vista muestra la disposición física de los distintos nodos que componen el Sistema Web *Jigsaw* y el reparto de los componentes sobre dichos nodos.

A continuación en la figura 15, se presenta el Diagrama que representa a esta Vista.

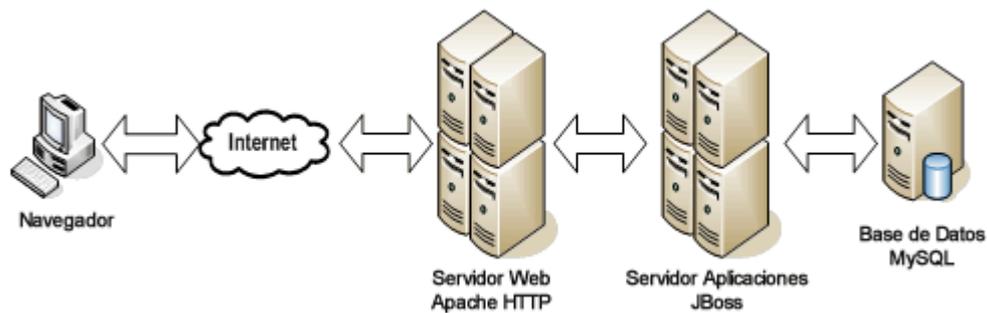


Figura 15: Diagrama de Despliegue

- **Navegador**
Es la interfaz que permite visualizar las páginas web.
- **Servidor Web Apache HTTP**
Es el responsable de aceptar las peticiones HTTP de los clientes (navegadores) y envía las respuestas con los datos solicitados a los mismos.
- **Servidor de Aplicaciones**
Servidor de Aplicaciones J2EE que tiene incluido el contenedor de *Servlets* *JBoss*.
- **Base de Datos MySQL**
Donde residen los objetos persistentes del sistema.

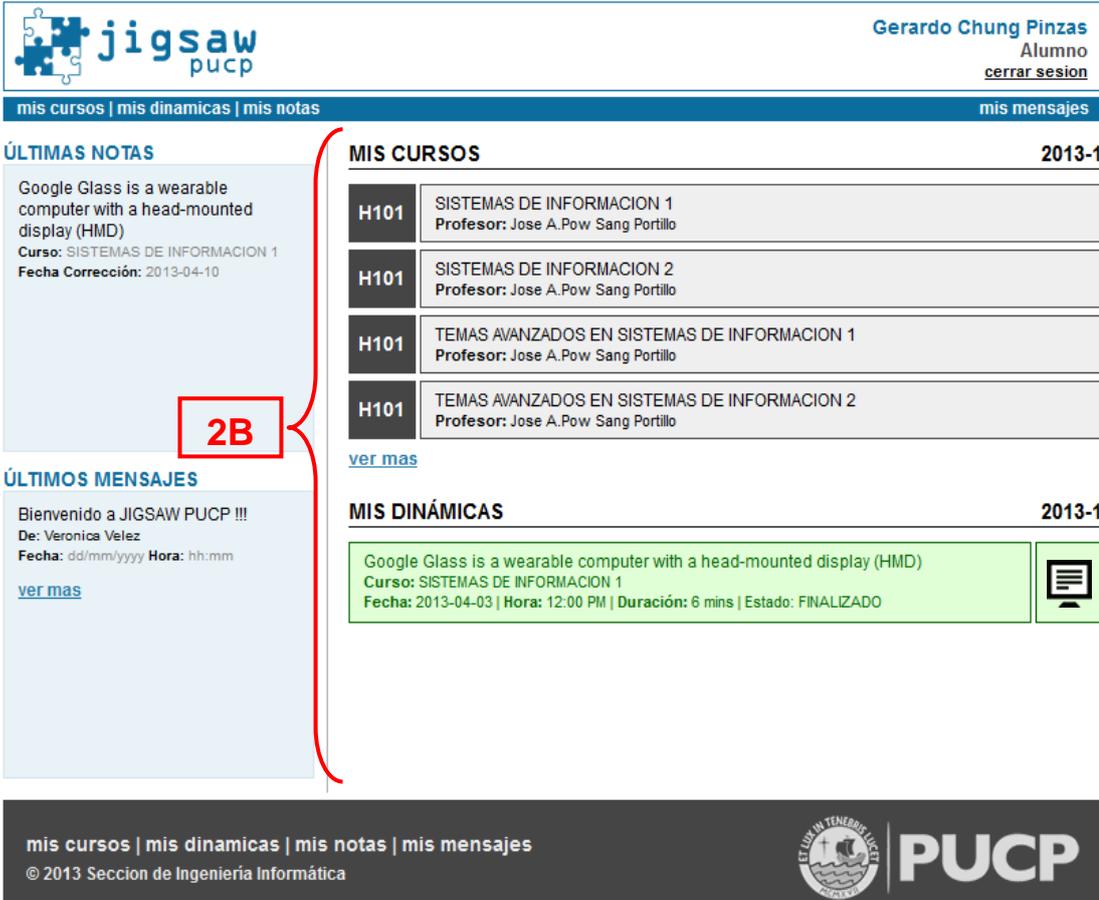
3.2. Diseño de Interfaz Gráfica

El diseño de la interfaz gráfica cumple un rol muy importante dentro de la presente tesis ya que es el intermediario entre el usuario y la lógica del sistema. Es por ello que se ha tomado especial cuidado en la navegabilidad, el acceso y

la distribución de cada de las secciones que forman parte de las pantallas. A continuación, se describirá las pantallas más representativas del sistema.

3.2.1. Pantalla Principal

A continuación en la Figura 16, se presenta el detalle de la interfaz de la pantalla principal del Sistema Web *Jigsaw*. Esta pantalla contiene un resumen de los datos principales de los cursos en los que se encuentra inscrito un Alumno. Así también, muestra un resumen de las actividades y evaluaciones programadas.



The screenshot shows the main interface of the Jigsaw PUCP system. It features a header with the user's name and a navigation bar. The main content area is divided into several sections: 'ÚLTIMAS NOTAS', 'ÚLTIMOS MENSAJES', 'MIS CURSOS', and 'MIS DINÁMICAS'. The footer contains navigation links and the PUCP logo.

1 Header area containing the Jigsaw PUCP logo and user information: Gerardo Chung Pinzas, Alumno, cerrar sesion.

2A Left sidebar containing 'ÚLTIMAS NOTAS' and 'ÚLTIMOS MENSAJES'.

2B 'MIS CURSOS' section for 2013-1, listing four courses:

MIS CURSOS		2013-1
H101	SISTEMAS DE INFORMACION 1 Profesor: Jose A.Pow Sang Portillo	
H101	SISTEMAS DE INFORMACION 2 Profesor: Jose A.Pow Sang Portillo	
H101	TEMAS AVANZADOS EN SISTEMAS DE INFORMACION 1 Profesor: Jose A.Pow Sang Portillo	
H101	TEMAS AVANZADOS EN SISTEMAS DE INFORMACION 2 Profesor: Jose A.Pow Sang Portillo	

3 Footer area containing navigation links (mis cursos | mis dinamicas | mis notas | mis mensajes), copyright information (© 2013 Seccion de Ingeniería Informática), and the PUCP logo.

Figura 16: Pantalla Principal

Esta sección organiza el contenido de lo que se va a presentar al inicio de la sesión del usuario que ingresa al Sistema. La información presentada en esta parte servirá como una guía de las diferentes funcionalidades que presenta el Sistema Web *Jigsaw*; y por tal motivo es necesario que esté ubicada en un lugar visible y que ocupe la mayor parte del área de la pantalla.

- **Sección 1**

La cabecera muestra la identificación del usuario, su nombre y el rol asignado para su sesión. Se presenta una lista de opciones para su configuración personal. Se eligió esta posición porque los datos presentados en esta parte son básicos para el desarrollo de la sesión y por tanto se requiere de la ubicación más visible para el usuario.

- **Sección 2A**

Mis Notas / Mis Evaluaciones: Se muestra los eventos más recientes programados para el usuario. Este puede variar según el perfil del usuario que ingresa al Sistema: para el Alumno se mostrará el evento “Mis Notas” y para el Docente se mostrará el Evento “Mis Evaluaciones”.

Mis Mensajes: Se muestra la mensajería instantánea más reciente recibida por el usuario. Cada mensaje presenta un enlace al contenido de este, así como otro enlace a los datos personales del remitente. Se eligió esta posición porque los datos presentados en esta parte buscan organizar la información producida de las sesiones *Jigsaw* del Alumno, por lo que se coloca para complementar la información académica del lado derecho.

- **Sección 2B**

Mis Cursos: Se muestra los datos y la información acerca de las sesiones de clase programadas de cada curso en la que el alumno se ha inscrito.

Mis Dinámicas: Se muestra los datos y la información acerca de las dinámicas programadas de cada curso en la que el alumno se ha inscrito.

- **Sección 3**

Esta sección delimita el tamaño que ocupa la información que se muestra por página. Además es un pie de página informativa acerca del producto. Se eligió esta posición porque se busca de una manera llamativa delimitar la información a mostrar en pantalla y así evitarle al usuario la necesidad de tener contenidos con *scrolls* demasiado amplios.

3.2.2. Pantalla del Módulo de Planificación

A continuación en la Figura 17, se presenta el detalle de la interfaz de la pantalla de Creación de una nueva Dinámica, perteneciente al Módulo de Planificación. Esta pantalla permite efectuar la creación de una sesión de clase *Jigsaw*.

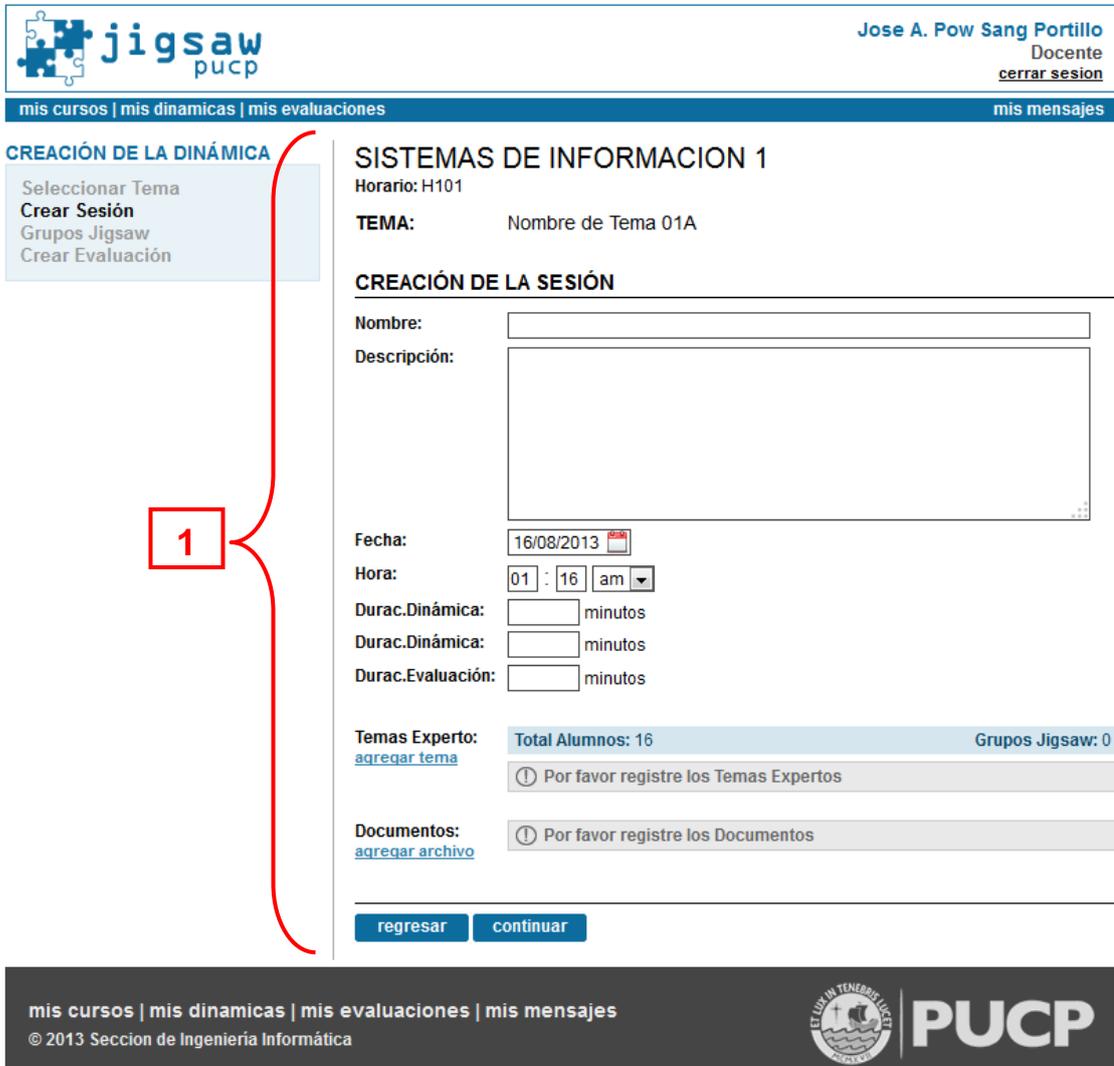


Figura 17: Pantalla Módulo de Planificación

Esta sección organiza el contenido de lo que se va a presentar en la Sesión de Clase *Jigsaw* (llamada también: Dinámica). Los datos presentados y registrados en esta parte son básicos para el desarrollo de la sesión y por tal motivo es necesario que estén ubicados en un lugar visible y que ocupen la mayor parte del área de la pantalla.

- **Sección 1**

Se presenta un identificador de la pantalla y las opciones referentes al manejo de la información presentada. A continuación se muestran los datos que están almacenados en el Sistema para la edición y/o modificación según sea la elección del usuario.

3.2.3. Pantalla del Módulo Ejecución

A continuación en la Figura 18, se presenta el detalle de la interfaz de la Herramienta Cooperativa para elaboración de los Casos de Uso, perteneciente al Módulo de Ejecución. Esta pantalla contiene un contador en la parte superior donde se monitoreará el cumplimiento de los tiempos asignados. Así mismo permite establecer una comunicación en tiempo real entre los participantes de la sesión a través de una herramienta de Chat.

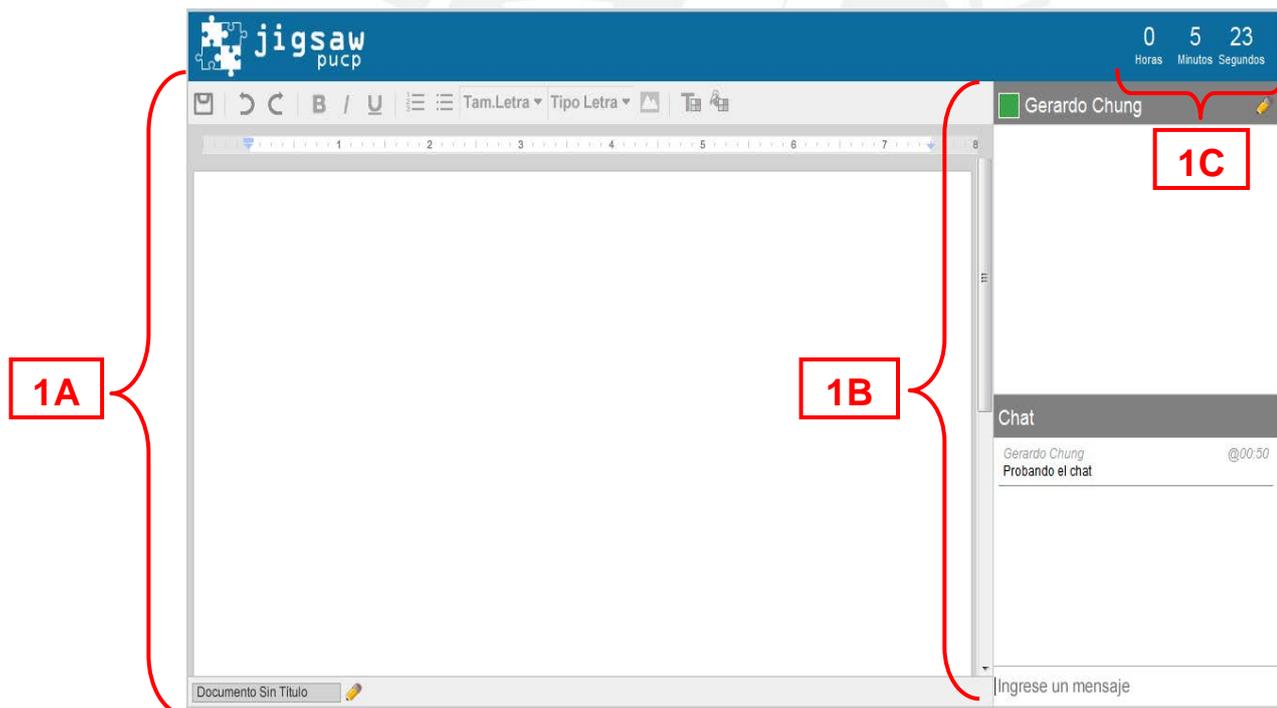


Figura 18: Pantalla Módulo de Ejecución

Esta distribución de pantalla organiza las comunicaciones entre los integrantes del grupo y presenta los diagramas que realizarán durante la ejecución de la sesión de clase.

- **Sección 1A**

En esta sección se presenta el espacio de diagramación personal que cada miembro de un grupo (*Jigsaw* o Experto) cuenta. Todo cambio realizado se efectúa en tiempo real y se refleja en las demás sesiones de los participantes.

- **Sección 1B**

En esta sección se presenta la información de los participantes según el grupo al que estén asignados durante la dinámica. También presenta un Chat para lograr mayor fluidez en las comunicaciones y coordinaciones durante las sesiones de clase.

- **Sección 1C**

Se presenta un contador que se encargará de monitorear el cumplimiento de los tiempos asignados para esa etapa de la dinámica definidos según la etapa de planificación.

3.2.4. Pantalla del Módulo Evaluación

A continuación en la Figura 19, se presenta el detalle de la interfaz de la pantalla de desarrollo de una Evaluación, perteneciente al Módulo de Evaluación. Esta presenta un listado con el número de preguntas que forman parte de la evaluación (que a su vez permite la navegación entre las mismas), además de mostrar el detalle de la pregunta junto con las opciones de respuesta disponibles.

1A

1B

1C

0 5 44
Horas Minutos Segundos

jigsaw pucp

LISTADO DE PREGUNTAS

Pregunta 01 (5 pts)
Tipo: Opc.Simple

Pregunta 02 (4 pts)
Tipo: Opc.Múltiple

Pregunta 03 (6 pts)
Tipo: Desarrollada

finalizar evaluación

SISTEMAS DE INFORMACION 1

Horario: H101

TEMA: Nombre de Tema 01A

DINAMICA: Google Glass is a wearable computer with a head-mounted display (HMD)

Enunciado Inicial:
An early adopter program named the Glass Explorer Program is available for developers and consumers to test Google Glass and gauge how people will want to use Glass. Entry into the Explorer Program was made available to the general public on February 20, 2013, and ended on February 27, 2013. The program stated that it was looking for "bold, creative individuals" who wanted to test the device. Those who wanted to apply were required to post a message on Google or Twitter consisting of 50 words or less, featuring the hashtag

Pregunta 01

The Explorer Edition is available in Charcoal, Tangerine, Shale, Cotton, and Sky colors.[32] Users issue voice commands by first saying "ok glass", then the command, or they can scroll through the options using a finger along the side of the device. The Explorer Edition has an interchangeable sunglasses accessory which twists on or off easily. Monthly updates to the Glass are planned after the program starts. (5pts)

1. Google Glass has the ability to take photos and record 720p HD video. While video is recording, a recording light is displayed above the eye, which is unnoticeable to the wearer.
2. Glass will utilize many already-existing Google applications, such as Google Now and Google Maps. The device will also be able to display the weather.
3. Multiple features of Glass can be seen in a product video released in February 2013:
4. In general, reception for Google Glass has been positive in the technology industry.

responder

Figura 19: Pantalla Módulo de Evaluación

Esta sección organiza la información y contenido de las evaluaciones que se realizan después de concluir la fase de Grupos *Jigsaw*.

- **Sección 1A**

En esta sección se presenta la lista de preguntas que conforman la evaluación. Una vez seleccionada una de las preguntas, el detalle se mostrará en la siguiente Sección 1B.

- **Sección 1B**

En esta sección se presenta el enunciado principal, detalle de la pregunta seleccionada y las opciones disponibles de respuesta disponibles.

- **Sección 1C**

Se presenta un contador que se encargará de monitorear el cumplimiento de los tiempos asignados para la evaluación definidos según la etapa de planificación.

Capítulo IV: Construcción

En el presente capítulo se presentarán las etapas finales del proyecto que corresponden al diseño y construcción del mismo.

Para detallar la construcción del Sistema Web *Jigsaw* se presentará la selección del lenguaje de programación y herramientas empleadas para el desarrollo de la presente tesis.

Luego se detallarán las especificaciones de diseño que se consideraron en el desarrollo del Sistema Web *Jigsaw*, así como la presentación del Plan de Pruebas empleado, con el cual se garantiza su correcto funcionamiento.

4.1. Construcción

Esta sección brinda una descripción acerca de los patrones y tecnologías que se van a utilizar para el desarrollo de la presente tesis.

Los patrones de programación son soluciones o procedimientos que son sugeridos para afrontar problemas específicos que se presentan durante el proceso del desarrollo de un Sistema

4.1.1. Patrones de Diseño

Un patrón de diseño es una solución general que se identifica en problemas comunes de diseño de software. Este no es un diseño final que pueda ser transformado directamente en código, sino que su descripción sirve como plantilla para solucionar problemas que pueden presentarse en muchas situaciones.

Singleton Pattern

Este patrón de diseño se empleó para poder controlar y optimizar la creación de instancias de clase de objetos, invocadas principalmente por los métodos de la lógica de negocio; buscando de esta manera limitar a lo esencial el uso de recursos y memoria por parte del Sistema.

Este patrón se implementa creando una clase con un método que crea una nueva instancia de la clase si es que una aún no existe. Si ya hay una instancia existente, simplemente retorna una referencia a dicho objeto. Para asegurarse

que el objeto no pueda ser instanciado de otra manera, el constructor de esta clase se hace *private* o *protected*.

Factory Method Pattern

Este patrón de diseño se empleó para poder gestionar de manera más rápida la creación de una buena cantidad de POJOs (*Plain Old Java Objects*) que se emplearon para la interactuar con los *Actions* (Acciones) definidas por la lógica de negocio del Sistema empleando *Struts 2*.

Este patrón se implementa para manejar el problema de la creación de objetos sin especificar la clase exacta de la cuál ha sido creada. Este patrón de diseño maneja este problema al definir un método por separado para la creación de objetos, cuyas subclases pueden sobrescribirse para especificar el tipo de objeto que se desea crear. Los métodos de Fabrica (*Factory*) son comunes en Frameworks donde las librerías necesitan crear tipos de objetos que puedan ser usados como subclases por otras aplicaciones.

Façade pattern

Este patrón de diseño se empleó para poder administrar las invocaciones a las funciones de la lógica de negocio que se encuentran en los módulos que conforman el Sistema. Esto permitió constituir una colección de APIs más consistentes que permitieron encapsular las dependencias de código internas.

Este patrón se implementa por medio de un objeto que provee un acceso simplificado a un gran bloque de código mediante el desarrollo de una capa de interfaz intermedia. Esto ayuda a simplificar las comunicaciones que hay entre los módulos y clases, que se pueden ir complicando cuando un sistema comienza a crecer en funcionalidades. Una *Façade* podría resumirse en la creación de una pequeña colección de clases que usan una sola clase para ser accedidas.

Proxy Pattern

Este patrón de diseño se empleó para poder brindar seguridad sobre las invocaciones del usuario hacia la información y las funciones de la lógica de negocio que se encuentran en los módulos que conforman el Sistema.

Este patrón tiene por objetivo definir un objeto que sea un "proxy", y por ende que actúe entre el cliente y el objeto a utilizar, controlando el acceso y realizando las tareas necesarias cada vez que se quiera acceder a este objeto. Típicamente, una instancia de un objeto es creado y múltiples objetos "proxy" son creados, los cuáles contienen una referencia al objeto original. Cualquier operación realizada en los *proxies* son enviados al objeto original. Una vez que todas las instancias de los objetos *proxies* terminan de ser utilizadas, el objeto original es retirado de la memoria.

4.1.2. Patrones Core J2EE

Con el tiempo y crecimiento de J2EE, se ha notado que ocurren problemas similares durante el desarrollo de estos proyectos. A raíz de esto han emergido soluciones similares para todos estos problemas. Cuando se ha visto un problema y una solución recurrente, se ha intentado identificar y documentar sus características usando la plantilla de patrón, es por ello que nacen específicamente los Patrones Core J2EE.

Front Controller

Este patrón sugiere el uso de un controlador como el punto inicial de contacto para manejar las peticiones. El Controlador, como su nombre lo indica, maneja el control de peticiones, incluyendo la invocación de los servicios de seguridad como la autenticación y autorización, delegar el procesamiento de negocio, controlar la elección de una vista apropiada, el manejo de errores, y el control de la selección de estrategias de creación de contenido.

Intercepting Filter

Este patrón sugiere la creación de filtros conectables para procesar servicios comunes de una forma estándar sin requerir cambios en el código principal del procesamiento de la petición. Los filtros interceptan las peticiones entrantes y las respuestas salientes, permitiendo un pre y post-procesamiento. Mediante esto se puede añadir y eliminar estos filtros a discreción, sin necesitar cambios en el código existente.

View Helper

Este patrón sugiere la implementación de una vista que contiene código de formateo, y que delega sus responsabilidades de procesamiento en sus clases

de ayuda, implementadas como *Java Beans* o etiquetas personalizadas. Las clases de ayuda o *helpers* también almacenan el modelo de datos intermedio de la vista y sirven como adaptadores de datos de negocio.

Session Façade

Este patrón sugiere el uso de un *bean* de sesión como una fachada (*facade*) para encapsular la complejidad de las interacciones entre los objetos de negocio participantes en un flujo de trabajo. El *Session Facade* maneja los objetos de negocio y proporciona un servicio de acceso uniforme a los clientes.

Transfer Object

Este patrón sugiere el uso de un *Transfer Object* para encapsular los datos de negocio. Se utiliza una única llamada a un método para enviar y recuperar el *Transfer Object* (por ejemplo: entre funciones y formularios). Cuando el cliente solicita los datos de negocio al *bean enterprise*, éste puede construir el *Transfer Object*, rellenarlo con sus valores de atributos y pasarlo por valor al cliente.

Data Access Object

Este patrón sugiere el uso de un *Data Access Object* (DAO) para abstraer y encapsular todos los accesos a la fuente de datos. El DAO maneja la conexión con la fuente de datos para obtener y almacenar datos, éste implementa el mecanismo de acceso requerido para trabajar con la fuente de datos. Los componentes de negocio que tratan con el DAO utilizan una interfaz simple expuesta por el DAO para sus clientes. El DAO oculta completamente los detalles de implementación de la fuente de datos a sus clientes. Como el interfaz expuesto por el DAO no cambia cuando cambia la implementación de la fuente de datos subyacente, este patrón permite al DAO adaptarse a diferentes esquemas de almacenamiento sin que esto afecte a sus clientes o componentes de negocio. Esencialmente, el DAO actúa como un adaptador entre el componente y la fuente de datos.

4.1.3. Tecnología

La tecnología que se empleó para el desarrollo de la presente tesis fue JAVA, debido a la gran portabilidad de sus aplicaciones, la gran comunidad de

desarrolladores que dan soporte a esta tecnología y principalmente porque es *open source* y no presenta gastos de licencias.

Framework J2EE – Apache Struts 2

Para una aplicación Web J2EE, el cliente envía información al servidor por medio de un formulario Web. Esta información es procesada por un *Servlet*, el cual se encarga de ejecutar la lógica de negocio y realizar el llamado a una Base de Datos, generando un respuesta por medio de una página HTML o una página JSP (Java Server Page) la cual encapsula HTML y código Java para obtener el mismo resultado. Estas aproximaciones son consideradas inadecuadas para proyectos grandes ya que la mezcla de la lógica con la presentación hace difícil su mantenimiento.

El objetivo de *Struts 2* es separar de manera clara el Modelo (la lógica de la aplicación que interactúa con la Base de Datos) de la Vista (HTML que se presenta al cliente) y el Controlador (la instancia que pasa la información entre la Vista y el Modelo).

A continuación en la Figura 20, se muestra un Diagrama detallando la Arquitectura de una aplicación que hace uso del *Framework Apache Struts 2*.

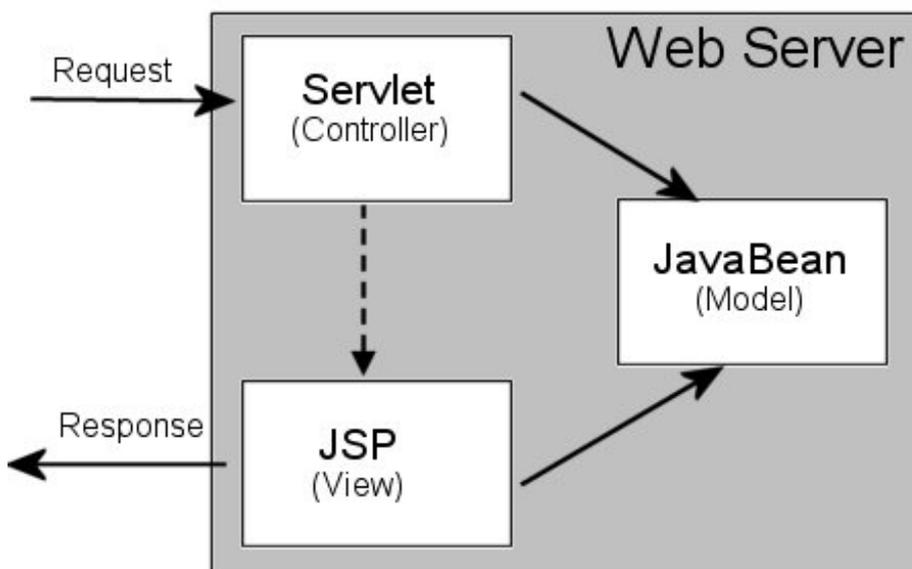


Figura 20: Arquitectura de una Aplicación Struts 2 [21]

En el caso de *Struts 2*, los pedidos del cliente llegan al servidor de aplicaciones y este los envía al *Servlet* que controla las peticiones recibidas. Este *Servlet*

analiza el pedido y revisa su configuración para ver qué hacer con este. Una vez que haya terminado la revisión del pedido, el *Servlet* decidirá qué Acción (*Action*) es la que realizara el procesamiento del pedido. Una Acción (*Action*) se encuentra incluida dentro de un Controlador (*Controller*). La labor de las acciones es la de obtener los parámetros y enviar los datos a los Objetos de Negocios (*Java Beans*) correspondientes. Finalmente la acción retorna una respuesta, que es una cadena de caracteres que indica cual es la página JSP a mostrar al usuario. La librería de *tags* de JSP permite la lectura y escritura de estos datos desde la capa de presentación sin la necesidad de usar de algún código Java en el código HTML.

La opción inicial para el desarrollo de la presente Tesis fue el *Framework Apache Struts (Struts 1)*; pero luego de algunas revisiones de los conceptos anteriormente presentados, se optó por el empleo del *Framework Apache Struts 2*. Ambos *frameworks* mantienen el mismo núcleo y arquitectura, presentándose mejoras más relevantes en los siguientes puntos.

- Independencia del API de *Servlet*: en el caso de *Struts 1*, las Acciones tienen independencias en los objetos *HttpServletRequest* y *HttpServletResponse*, ya que estos son pasados al método "execute" cuando la acción es invocada. En el caso de *Struts 2*, las acciones son simples POJOs (*Plain Old Java Objects*) que no tienen dependencia y pueden ser probados por separado.
- Programación de Clases Abstractas: en el caso de *Struts 1*, las acciones necesitan heredar (*extends*) de una clase base abstracta para realizar todas sus funcionalidades. En el caso de *Struts 2* esto se simplifica puesto que estos son simple POJOs. Cualquier POJO que tenga un método "execute" puede actuar como una Acción en *Struts 2*.
- Validación de Contenidos: en *Struts 1* la validación de contenidos se hace en el *Action Form*. En caso de *Struts 2*, este soporta este tipo de validación, además de la validación manual y la validación del *framework XWork*.
- Captura de Datos: en el caso de *Struts 1*, éste utiliza un *Action Form* para obtener los datos del pedido. Todos los *Action Form* deben heredar de una clase dependiente del *Framework*, por lo que se crean clases redundantes para la captura de datos. En el caso de *Struts*, este emplea *Action*

Properties que pueden ser accedidas desde la página JSP por medio de librerías de *tags*, además de soportar formularios tipo POJOs.

- **Mostrar Datos:** en el caso de *Struts 1*, se usa el mecanismo estándar JSP para mostrar objetos en la página que se desea acceder. En el caso de *Struts 2*, este usa una Tecnología "*Value Stack*", que permite acceder a las librerías de *tags* sin modificar o agregar comodines al código fuente de la página a utilizar.
- **Control de la Ejecución de las Acciones:** en el caso de *Struts 1*, éste soporta pedidos de procesamiento de acciones por separado, pero todas estas deben compartir el mismo ciclo de vida. En el caso de *Struts 2*, este soporta la creación de diferentes ciclos de vida en cada Acción empleada, lo que permite mayor independencia en su desarrollo y empleo.

Librerías AJAX: JQuery / DojoToolkit

AJAX (*Asynchronous JavaScript and XML*) es una técnica de desarrollo para la creación de aplicaciones Web interactivas. La premisa general es hacer que las páginas sean más dinámicas al mostrar su información, mediante el intercambio de pequeñas cantidades de datos con el servidor como procesos background [25]. Mediante esto se trata de incrementar la interactividad, velocidad y funcionalidad de la página Web. AJAX es asíncrono en el hecho que no interfiere con la normal carga de la página. Se emplea el lenguaje de programación JavaScript para hacer las llamadas a funciones. El manejo de datos se formatea generalmente usando XML, como se refleja en el nombre del Objeto *XMLHttpRequest* en la que AJAX está derivado

jQuery es una biblioteca de JavaScript creada inicialmente por John Resig y es considerado actualmente el estándar para el manejo simplificado de interacciones AJAX en los desarrollos Web [26]. Esta biblioteca permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Dojo ToolKit es un *framework* creado inicialmente por Alex Russell, que contiene APIs y *widgets* (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX [27]. Contiene un sistema de empaquetado inteligente, los efectos de UI, *drag and drop APIs*, *widget APIs*, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX. Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas (*bookmarking*), y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos.

Entre las ventajas que presenta la Tecnología AJAX con respecto al manejo de datos se pueden presentar:

- Es *open source* y está basado en estándares abiertos soportados por varios Browsers y plataformas: la mayoría de estas tecnologías que se emplean han sido utilizadas con el transcurso de los años: Javascript, XML, HTML, CSS y *XML HTTP Request Object*.
- AJAX permite un mayor énfasis al rol de la experiencia del usuario con la aplicación, mediante la eliminación del problema de cargar toda la información de las pantallas y su lenta respuesta, situaciones que han existido desde la concepción de Aplicaciones Web.
- Se reduce el uso de ancho de banda en las aplicaciones Web, debido a la transferencia de porciones pequeñas de datos, permitiendo además tiempos de respuesta menores y haciendo óptima la aplicación.

Entre los beneficios que presenta *jQuery* están:

- La mayor ventaja del uso de una librería como *jQuery* para trabajar con AJAX reside principalmente en la gran cantidad de código que se puede ahorrar.
- Gracias a *jQuery* se puede reducir el código *JavaScript* que se necesita para alterar elementos en una página web, realizar animaciones, cargar contenidos en AJAX (haciendo peticiones tanto GET como POST), cambiar estilos CSS -y en la mayoría de las ocasiones- solo será requerido a lo mucho una línea de código.

- Otra de las ventajas que se obtiene del uso de *jQuery* en lugar de escribir código JavaScript sin ayuda, es que se obtiene mayor compatibilidad con los navegadores, ya que el equipo de desarrollo de *jQuery* ya ha tenido en cuenta las distintas sintaxis y particularidades de cada navegador.

Entre los beneficios que presenta *Dojo Toolkit* están:

- Ajax en Tags: abstrae los componentes Ajax en Tags de fácil uso. Esto permite inicializar los componentes Ajax de manera más fácil y práctica: `<div dojoType="RichTextEditor"></div>`.
- Normaliza los APIs de *Javascript*: presenta sólo un modo de *widget* que abstrae otras APIs y provee un Standard de representación, documentación y manipulación de *widgets*. En otras palabras se puede crear *widgets* reutilizables de varias librerías donde se puede elegir cual es la mejor que cumple los requisitos.
- Estandarización de Modelos de datos: provee un modelo de datos común que permite varias estructuras como Árboles, Vistas y otras descritas en el formato de Notación de Objetos *JavaScript* (JSON), con esto los *Widgets* de otras plataformas pueden ser usados con los mismos datos.
- Comunicación con el exterior: *Javascript* no puede acceder directamente a contenido fuera del dominio usando *XMLHttpRequests* debido a la seguridad que presentan los browsers.

Servidor de Aplicaciones JBoss

Un servidor de aplicaciones es un Servidor en una red de computadoras dedicada a correr aplicaciones. Para este punto se ha optado por el uso del servidor de Aplicaciones *JBoss*, el cual es un servidor de Aplicaciones J2EE que tiene incluido el Servidor *Tomcat*.

Tomcat es un contenedor que provee un *Framework* para desplegar y ejecutar *Servlets* (incluidos JSP) en un servidor Web. Con una definición limitada de un Servidor de Aplicaciones, se podría decir que *Tomcat* es en sí un Servidor de Aplicaciones, debido a que puedes construir aplicaciones completas usando solamente *servlets*. Pero, eso sería muy restrictivo, pues en la actualidad los servidores de aplicaciones deben manejar las transacciones y las conexiones a

Base de Datos de manera transparente, algo que no se aplica en el caso de *Tomcat*.

A continuación en la Figura 21 se muestra un diagrama detallando la arquitectura del Servidor de Aplicaciones *JBoss*.

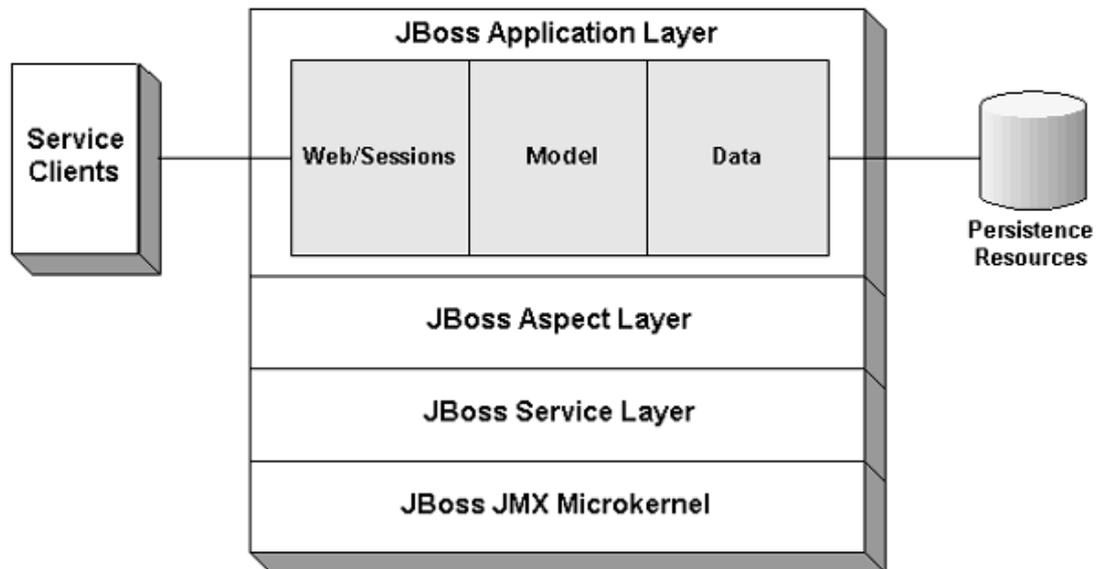


Figura 21: Arquitectura Servidor JBoss [22]

JBoss puede soportar cualquier JVM y puede correr en numerosos sistemas operativos incluido Windows, mientras una correcta JVM (Java Virtual Machine) está presente.

JBoss es un programa *open source* (sin costo de licencia), fácil de manejar y administrar, que posee una gran compatibilidad y complemento con la herramienta de mapeo objeto-relacional JPA. Por ello se podría constituirse como una opción confiable y económica frente a otros Servidores de Aplicaciones, tales como *BEA Weblogic* e *IBM Websphere*, que son de software propietario y que demandan mayor soporte para la realización de sus procesos.

Base de Datos MySQL[30]

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder y procesar datos

guardados en un computador, se necesita de un software de administración como el *MySQL* Server.

MySQL emplea una administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

Las ventajas que presenta la base de datos *MySQL* son:

- Es que es un software *open source* (sin costo de licencia).
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Es fácil de manejar y administrar; además de poseer mayor documentación y soporte técnico en comparación con otras soluciones de software libre (como por ejemplo: *PostgreSQL*).

A continuación en la Figura 22, se muestra un diagrama detallando la arquitectura del Sistema de Administración de Base de datos *MySQL*.

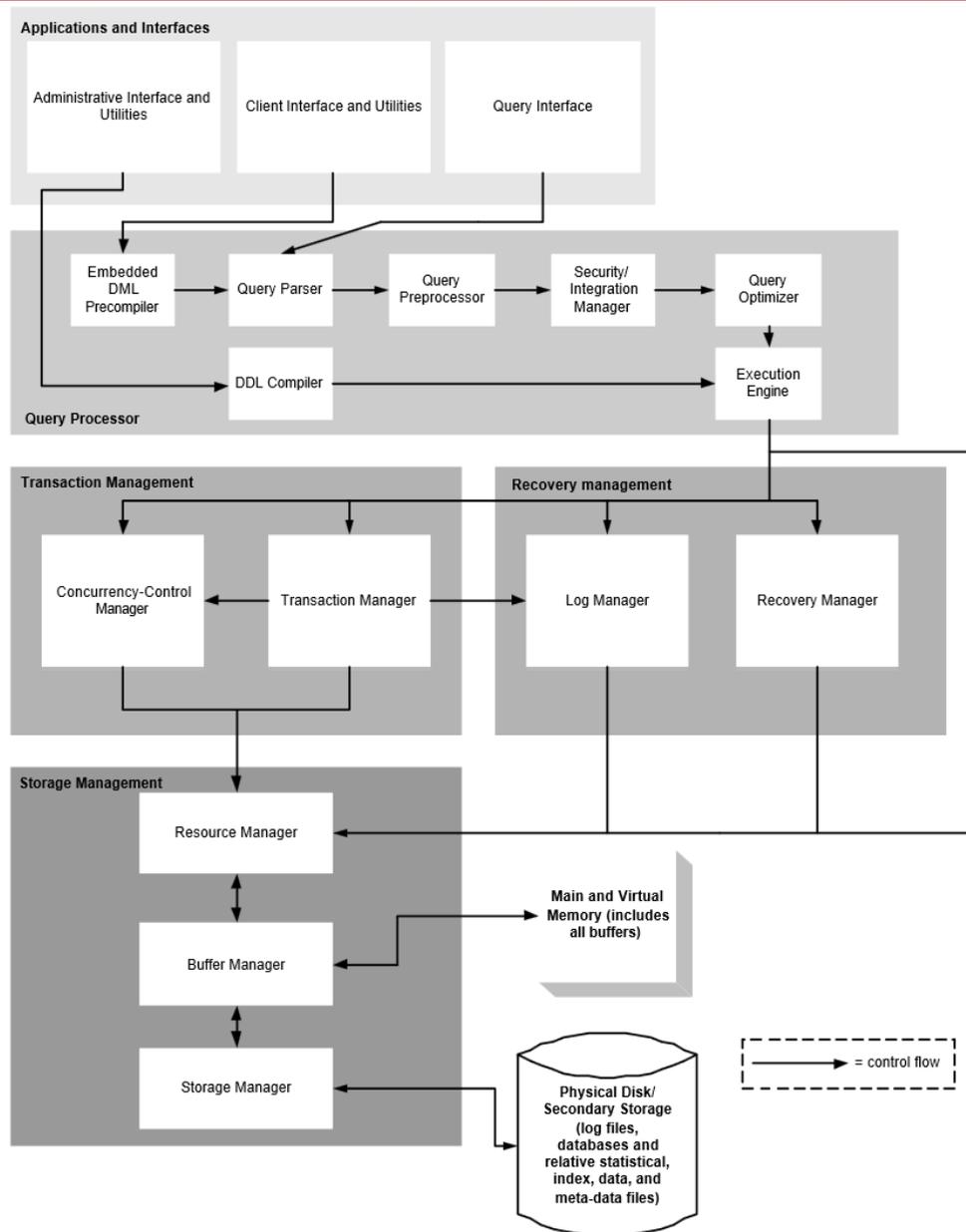


Figura 22: Arquitectura MySQL [23]

Librería para el mapeo MyBatis

El mapeo objeto-relacional (más conocido por su nombre en inglés, *Object-Relational mapping*, o sus siglas O/RM, ORM, y *O/R mapping*) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto

posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

A diferencia de las herramientas ORM, *MyBATIS* no mapea objetos Java a tablas de base de datos sino métodos a sentencias SQL. Permite utilizar todas las funcionalidades de la base de datos como procedimientos almacenados, vistas, consultas de cualquier complejidad o funcionalidades específicas del proveedor [24].

Simplifica la programación frente al uso directo de JDBC. Las líneas de código necesarias para ejecutar una sentencia se reducen casi siempre a una. Esta simplificación ahorra tiempo y evita errores habituales como olvidar cerrar una conexión a base de datos, realizar incorrectamente un mapeo de datos, exceder el tamaño de un *ResultSet* u obtener varios resultados cuando se esperaba sólo uno [24].

Proporciona un motor de mapeo de resultados SQL a árboles de objetos basado en información declarativa, además de soportar la composición de sentencias SQL dinámicas mediante un lenguaje con sintaxis tipo XML.

Entre las ventajas que presenta la librería *MyBATIS* con respecto al manejo de datos se presentan:

- Requiere menos clases e interfaces.
- Virtualmente elimina el desarrollo de descriptores mediante anotaciones.
- Direcciona la mayoría de especificaciones con las anotaciones por defecto.
- Provee una forma más limpia, fácil y estandarizada de mapear objetos relacionales.
- Elimina la necesidad de desarrollar código de búsqueda.
- Otorga soporte a herencia, polimorfismo y *queries* polimórficos.
- Otorga soporte a *queries* dinámicos y estáticos.
- Añade el lenguaje de *queries* para persistencia en JAVA.

Para esta librería, la curva de aprendizaje es corta; los tiempos de desarrollo, líneas de código y archivos que se necesitan para que la aplicación funcione se

reducen considerablemente frente a otros ORM conocidos (por ejemplo: *Hibernate*).

Como acotación a la conclusión anterior el modelo de POJO's en el que se basa *MyBATIS* hace que sea posible tener aplicaciones livianas (*lightheight applications*).

APIs de Concurrencia - Open Cooperative Web Framework

Open Cooperative Web Framework permite el manejo de interacciones en tiempo real entre usuarios remotos y origen de datos externos por medio de librerías *Javascript*. Este tipo de aplicaciones emergentes son referidas como Aplicaciones Web Cooperativas (*Cooperative Web Applications*). Este Framework maneja las notificaciones remotas de cambios realizados por el usuario, la resolución de conflictos al momento de realizar actualizaciones de manera concurrente y la convergencia en el estado de la aplicación utilizando algoritmos de Transformación Operacional. El objetivo de este *Framework* es el de brindar soporte cooperativo entre todos los browsers y aparatos [28].

Este *Framework* en su continua evolución y construcción de versiones busca servir como un medio de construcción para los desarrollos web que buscan ser empleados por un solo usuario o simultáneamente por un grupo de usuarios remotos [28].

Las ventajas que nos presenta el *Open Cooperative Web Framework* son:

- Es que es un software *open source* (sin costo de licencia).
- Las librerías encapsulan de manera efectiva la complejidad de los algoritmos de concurrencia, dejando al usuario únicamente la necesidad de invocar sus funciones.
- Permite la creación de herramientas cooperativas mediante el empleo del Framework *AJAX Dojo Toolkit*.

4.1.4. Estándares de Programación

Un Estándar de Programación es un término que describe convenciones para escribir código fuente en ciertos lenguajes de programación. Para el caso de la presente Tesis se emplearon las siguientes:

Notación de Clases

Las clases que se desarrollaron y emplearon la arquitectura presentada por el *framework Struts 2*, deberán seguir la siguiente sintaxis:

Do<Sigla_de_Modulo><nombre_de_la_accion>.java

Se identifica a una clase que representa una Acción (*Action*) que se realiza ante una solicitud de la Vista (*View*).

<nombre_entidad>Controller.java

Se identifica a los controladores (*Controlllers*) que contienen la lógica de negocio una entidad que participa de un módulo del Sistema.

<nombre_entidad>Model.java

Se identifica a las clases que gestionan la información del modelo de datos (*Model*).

<nombre_entidad>Mapper.java

Se identifica una clase que sirve de interfaz con la base de datos y la ejecución de la librería MyBATIS.

<nombre_entidad>Bean.java

Se identifica a un POJO (Plain Old Java Object) que contiene la información a emplear dentro de una lógica de negocio que se ejecuta en el Sistema.

Notación de Procedimientos

Los nombres de los procedimientos deben de llevar la primera palabra como verbo en minúscula que indique la acción a realizar, y el resto de palabras unidas con la primera letra en mayúsculas.

El procedimiento deberá tener la siguiente sintaxis:

<verbo_accion><detalle_de_identificadores>

Notación de Variables

Las variables dependiendo de su alcance se dividen en globales y locales, el nombre de la variable consta de un prefijo seguido del nombre de la variable definido por el consultor. Para el uso de separadores o espacios en blanco, se empleará un “_” (*underscore*).

La variable deberá tener la siguiente sintaxis:

<ubicación><tipo de dato>_<nombre de la variable>

A continuación en el Cuadro 11 se detalla los prefijos de ubicación empleados.

NOMBRE	PREFIJO
Global	g
Local	l

Cuadro 11: Prefijos para notación de ubicación de variables

A continuación en el Cuadro 12 se detalla los prefijos de tipo de dato empleados.

NOMBRE	PREFIJO
String	s
Char	c
Boolean	b
Numérico	n
Integer	i
List	lst
Array	arr
Objeto	obj
DateTime	dt
Collection	cl

Cuadro 12: Prefijos para notación de tipo de dato de variables

Notación de Controles Visuales

Los controles visuales deben poseer un nombre estándar, el cual se identificara con un prefijo todo en minúscula, seguido por el nombre del control.

La notación de un Control Visual deberá tener la siguiente sintaxis:

<prefijo>_<nombre_del_control>

A continuación en el Cuadro 13 se detalla los prefijos de los Controles empleados.

NOMBRE	PREFIJO
Label	lb
TextBox	txt
ComboBox	ddl
ListBox	lst
DialogBox	dlg
Option	op
Checkbox	ch
Radio Button	rd

Cuadro 13: Prefijos para notación de controles visuales

4.2. Pruebas

Esta sección provee una descripción acerca de la organización del proceso de pruebas que se realizaron en la presente tesis.

4.2.1. Estrategia de Pruebas

Los tipos de prueba que se detallarán a continuación son:

- Pruebas de Aceptación: cuyo propósito es el confirmar que el sistema satisface los requerimientos del negocio y provee un nivel de confianza de que el sistema trabaja adecuadamente e indicará que todos los requerimientos de sistema fueron implementados correctamente.
- Pruebas de Seguridad: cuyo propósito es comprobar las funcionalidades de validación y mantenimiento de usuarios así como la verificación de la asignación de perfiles, dinámicas, evaluaciones, accesos a entornos cooperativos; y el mantenimiento de los mismos.

4.2.2. Pruebas de Aceptación

Para la presente sección se presentarán las pruebas para un Caso de Uso de cada módulo que conforma el Sistema (Planificación, Ejecución y Evaluación). Estas verificarán la correcta implementación de los flujos básicos y alternativos de todos los casos de uso a implementar. Confirmando lo propuesto en el punto anterior.

Para mayor detalle de las pruebas realizadas a todos los Casos de Uso que conforman el Sistema, referirse al “Anexo IV – Plan de Pruebas”.

A. Módulo de Planificación

A continuación se describirá el detalle de la prueba de aceptación del Caso de Uso “Consultar Información del Curso”, el cual permite al Usuario consultar la información de un curso.

Objetivo Prueba:	Probar el funcionamiento del flujo básico del Caso de Uso “Consultar Información del Curso”.
Precondición:	El Usuario ha sido admitido en el Sistema
Descripción de la prueba:	<ul style="list-style-type: none"> • El Usuario elegirá un curso de la lista que presente el Sistema.

	<ul style="list-style-type: none"> El Usuario seleccionará uno de los Temas que conforman el curso, que presenta el Sistema mediante una lista.
Resultados Esperados:	<ul style="list-style-type: none"> Se mostrará el detalle del Curso: la lista de horarios con sus respectivos profesores y la lista de temas que conforman el curso Se mostrará en pantalla el detalle del Tema seleccionado: Nombre del Tema y la Descripción del Tema.

B. Módulo de Ejecución

A continuación se describirá el detalle de la prueba de aceptación del Caso de Uso “Ingresar a Sesión Cooperativa”, el cual permite al Alumno ingresar a la Sesión Cooperativa de una Dinámica asignada.

Objetivo Prueba:	Probar el funcionamiento del flujo básico del Caso de Uso “Ingresar a Sesión Cooperativa”.
Precondición:	El Alumno ha sido admitido en el Sistema
Descripción de la prueba:	<p>SELECCIONAR DINÁMICA</p> <ul style="list-style-type: none"> El Usuario seleccionará una de las Dinámicas asignadas y que el Sistema presenta por medio de una Lista. <p>DETALLE DE LA DINAMICA</p> <ul style="list-style-type: none"> El Usuario seleccionará el estado que esté "En Proceso" del menú que indica los estados de la Dinámica.
Resultados Esperados:	<p>SELECCIONAR DINÁMICA</p> <ul style="list-style-type: none"> El Sistema mostrará en pantalla el Menú de Estados de la Dinámica, Nombre del Curso, Horario del Curso, Tema del Curso, Nombre de la Dinámica, Descripción de la Dinámica, Título y Descripción del Tema Experto Asignado y la documentación asignada a la Dinámica Elegida (Título del Documento, Descripción del

	<p>Documento y el Nombre del Archivo referente al Documento)</p> <p>DETALLE DE LA DINAMICA</p> <ul style="list-style-type: none"> • El Sistema mostrará en una nueva ventana la Herramienta Cooperativa para elaboración de los Casos de Uso.
--	---

C. Módulo de Evaluación

A continuación se describirá el detalle de la prueba de aceptación del Caso de Uso “Consultar Configuración de Evaluación”, el cual permite al Docente revisar la configuración de una Evaluación creada previamente.

Objetivo Prueba:	Probar el funcionamiento del flujo básico del Caso de Uso “Consultar Configuración de Evaluación”.
Precondición:	El Docente ha sido admitido en el Sistema.
Descripción de la prueba:	<p>SELECCIONAR EVALUACION</p> <ul style="list-style-type: none"> • El Usuario seleccionará una de las evaluaciones de la lista que presenta el Sistema. <p>DETALLE DE LA EVALUACION</p> <ul style="list-style-type: none"> • El Usuario seleccionará cada uno de los Hipervínculos asignados a cada pregunta que forma parte de la Evaluación.
Resultados Esperados:	<p>SELECCIONAR EVALUACION</p> <ul style="list-style-type: none"> • El Sistema mostrará el detalle de la Evaluación: una lista con las preguntas que forman parte la evaluación (cada una de ellas serán hipervínculos que actualizarán el contenido del lado derecho de la pantalla según la pregunta seleccionada); y la calificación asignada a cada pregunta. Al lado derecho mostrará el Nombre del Curso, Horario del Curso, Nombre del Tema del Curso, Nombre de la

	<p>Dinámica, Enunciado Inicial de la Evaluación, Enunciado de la pregunta seleccionada y una lista de respuestas disponibles.</p> <ul style="list-style-type: none"> • El Sistema indicará de manera visual cuál de las respuestas presentadas son las que han sido asignadas como correctas. <p>DETALLE DE LA EVALUACION</p> <ul style="list-style-type: none"> • El Sistema actualizará el contenido del lado derecho del detalle de la evaluación con la información referente a la pregunta seleccionada.
--	--

4.2.3. Pruebas de Seguridad

Las pruebas de seguridad que se realizaron fueron las siguientes:

- Verificar el control de acceso al sistema solo a usuarios validados
- Verificar el ingreso a una página visitada anteriormente dentro de sesión una vez que se haya cerrado la Sesión.
- Verificar que los alumnos matriculados un curso se les haya asignado una dinámica creada para ese curso.
- Verificar que los alumnos asignados a una Dinámica tengan acceso a las etapas de Sesión Cooperativa.
- Verificar que los Alumnos tengan asignada una evaluación concluida las Etapas de Sesión Cooperativa.

Capítulo V: Conclusiones y Trabajos Futuros

En este capítulo final, se presentarán los puntos a los que se concluye después de haber realizado las fases de captura de información, análisis y diseño del sistema. En la primera sección se listarán las conclusiones del proyecto. Finalmente, se presentarán las recomendaciones y propuestas a tomar en cuenta para futuros trabajos que se desarrollen en base a la elaboración del Sistema Web *Jigsaw*.

5.1. Conclusiones

La presente tesis tuvo como objetivo el analizar, desarrollar e implementar un Sistema Web que permita simplificar los procesos de implementación de la Técnica de Aprendizaje Cooperativo de Rompecabezas; permitiendo de esta manera su aplicación en la enseñanza de Desarrollo de Software, enfocándose específicamente en la elaboración cooperativa de Casos de Uso.

Al hacer referencia a la simplificación de los procesos de implementación de esta técnica, principalmente se buscaba prescindir de la necesidad de un ambiente físico para el desarrollo de la Técnica, de la necesidad de personal adicional para ejecutar las tareas de la sesión, y de la gestión e impresión de materiales físicos.

A continuación se presenta un listado con las conclusiones a las que se ha llegado al final de la presente tesis:

- El análisis y diseño de un Sistema Web que implementa la Técnica de Aprendizaje Cooperativo de Rompecabezas, brindó una opción para poder prescindir del empleo de un ambiente físico para el desarrollo de esta Técnica:
 - Mediante el empleo de un ambiente virtual configurado en el Sistema Web, se permite la interacción de los alumnos durante una Sesión Cooperativa en las etapas de Grupos *Jigsaw* y Grupos Expertos.
 - El empleo de un Sistema Web permitirá brindar un apoyo para lograr una mayor participación de los Alumnos y Docentes a una dinámica, puesto que al ser un Sistema que funciona a través de internet, este no

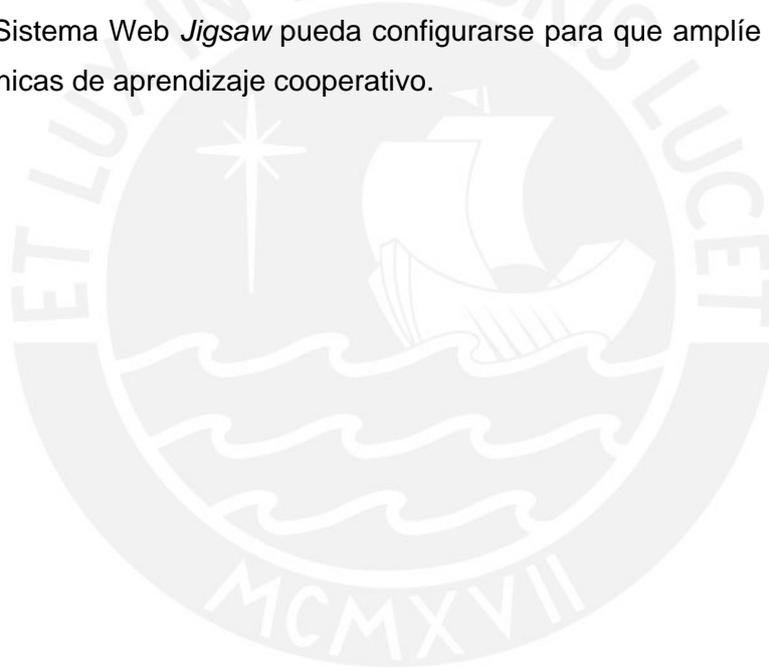
presenta restricciones geográficas o de tiempo entre los participantes de la Técnica.

- La elaboración de una Herramienta que permitió gestionar la participación de los alumnos durante una Sesión de la Técnica de Aprendizaje Cooperativo de Rompecabezas, podrá sugerir la disminución de la necesidad de solicitar personal adicional para ejecutar las tareas de esta técnica; ya que las tareas de las fases de ejecución (acerca de la administración de la participación de los alumnos y la conformación de los grupos dentro de la sesión) son ahora realizadas automáticamente por esta herramienta:
 - Con esta herramienta cooperativa, los Alumnos podrán participar en la totalidad de las fases de ejecución de la Técnica de Aprendizaje Cooperativo de Rompecabezas como si se tratara de un salón de clases, puesto que la herramienta agrupará a sus integrantes y administrará las reglas de participación según sea una sesión de Grupos Expertos o Grupos *Jigsaw*.
 - Dentro de las sesiones de Grupos Expertos y Grupos *Jigsaw*, los Alumnos podrán interactuar remotamente y brindar sus aportes de manera Cooperativa a la elaboración del Caso de Uso asignado por medio de esta herramienta; simulando las interacciones como si los alumnos estuviera presencialmente en un salón de clases.
- La implementación de la gestión y uso de documentación virtual dentro del Sistema, brindó una opción para prescindir de la necesidad de generación de material físico necesario para la realización de la Técnica de Aprendizaje Cooperativo de Rompecabezas dentro en un salón de clases. Esto se debe a que esta documentación ahora se encuentra disponible dentro de este ambiente virtual y puede ser revisada mediante sus visores de contenido disponibles.

5.2. Trabajos futuros

Algunas de las recomendaciones a considerar como trabajos futuros se muestran a continuación:

- Se recomienda la ampliación del módulo de Ejecución para que pueda abarcar una mayor cantidad de los diagramas disponibles en UML.
- Se recomienda una ampliación del módulo de Ejecución para que pueda permitir a los usuarios la creación de nuevas reglas para los diagramas que se empleen para diagramación.
- Se recomienda que la Herramienta Cooperativa para elaboración de los Casos de Uso pueda ser empleada por browsers multiplataforma. Debido a restricciones de la versión del *Framework OpenCowebe* (v1.0.1) con la que se desarrolló la presente tesis; las funciones cooperativas se visualizan con mayor precisión cuando se emplea un solo tipo de browser para todas las estaciones de usuario.
- El Sistema Web *Jigsaw* pueda configurarse para que amplíe su uso a otras técnicas de aprendizaje cooperativo.



Bibliografía y Referencias

- [1] Pow-Sang Portillo, J. A. (2006). *Aplicación del Método del Rompecabezas a una Sesión para el Aprendizaje de Diagramas de Clases de Análisis*. Lima: Pontificia Universidad Católica del Perú.
- [2] Aronson, E., & Patnoe, S. (1997). *The jigsaw classroom: Building cooperation in the classroom*, New York: Longman.
- [3] Slavin, R (1992). *Aprendizaje Cooperativo*. Barcelona: C. Rogers y P.Kutnich.
- [4] Pujolas, P & Lago, J.R. (2009) *Programa CA/AC (Cooperar para aprender, aprenderá cooperar). Algunas actuaciones para implementar el aprendizaje cooperativo*. Barcelona: Universitat de Vic. Facultat d'Educació.
- [5] Kagan, S. (1994). *Cooperative Learning*. California: Kagan Cooperative Learning
- [6] Johnson, D. W. & Johnson, R. T. (1991) *Cooperative learning lesson structures*. Edina, M.N: Interaction Books
- [7] Slavin, R. E. (1995). *Cooperative learning: Theory, research and practice* (2nd ed) Boston: Allyn and Bacon.
- [8] ZOPP in Brief, ZOPP Flipcharts (1987). *An Introduction to the Method*. Frankfurt: Deutsche Gesellschaft für Technische Zusammenarbeit.
- [9] Johnson, D. & Johnson, R. & Holubec, E. (1998): *Cooperation in the classroom* (7th ed). New York: Interaction Book Company.
- [10] Galardo, T. (2003), *Supporting JIGSAW-type Collaborative Learning*. Santiago: Universidad de Chile.
- [11] Cobos C, &, Salazar J. (2005) *Ambiente Computacional Para La Edición Cooperativa De Clases Con UML*. Cali: CLEI 2005.
- [12] Soller, A. (2002). *Workshop Designing Computational Models of Collaborative Learning Interaction*. Pensilvania: University of Pittsburgh.

- [13] Soller, A. (1998). *Promoting Effective Peer Interaction in an Intelligent Collaborative Learning System*. In *Proceedings of 4th International Conference on Intelligent Tutoring System*. Pensilvania: University of Pittsburgh.
- [14] Soller, A. (1999). *What makes peer interaction effective? Modeling effective communication in an intelligent CSCL*. American Association for Artificial Intelligence. Pensilvania: University of Pittsburgh.
- [15] Koch, M. (1995) *The Collaborative Multi-User Editor Project Iris*. Inst. für Informatik: Technische Univ.München.
- [16] Kruchten, P (2003). *The Rational Unified Process: An Introduction (3rd Ed)*. New York: Addison-Wesley Professional
- [17] Parker, J. (2012) "Jigsaw Strategy sheet #3".
<http://ejlazaros.iweb.bsu.edu/itedu691/pdf/Jigsaw.pdf>
- [18] Mahillah, S. (2012) "The Effectiveness of Teaching Reading Comprehension by Using Jigsaw Technique"
<http://smahillah.hubpages.com/hub/mamaazaputri2>
- [19] Google Docs - Wikipedia La Enciclopedia Libre
(http://en.wikipedia.org/wiki/Google_Docs)
- [20] Pontificia Universidad Católica del Perú - Campus Virtual
(<http://campusvirtual.pucp.edu.pe>)
- [21] Struts 2 - Why we need a controller architecture
(<http://www.javaranh.com/journal/2002/03/struts.html>)
- [22] Novell - Multi-tiered Service-Oriented Systems with JBoss
(<http://www.novell.com/coolsolutions/feature/16192.html>)
- [23] MySQL - Architecture Application Layer
(<http://linuxapachephpmysql.blogspot.com/2010/09/mysql-architecture.html>)
- [24] MyBatis - Wikipedia La Enciclopedia Libre
(<http://es.wikipedia.org/wiki/MyBatis>)
- [25] AJAX - Wikipedia La Enciclopedia Libre
(<http://es.wikipedia.org/wiki/AJAX>)

[26] JQuery - Wikipedia La Enciclopedia Libre

(<http://es.wikipedia.org/wiki/Jquery>)

[27] Dojo Toolkit - Wikipedia La Enciclopedia Libre

(http://es.wikipedia.org/wiki/Dojo_toolkit)

[28] Open Cooperative Web Framework - Project Overview

(<http://opencoweb.org>)

[29] Web Socket - Wikipedia La Enciclopedia Libre

(<http://es.wikipedia.org/wiki/WebSocket>)

[30] MySQL - Wikipedia La Enciclopedia Libre

(<http://es.wikipedia.org/wiki/MySQL>)

