



Sistema Gestor de Aprendizaje para el Diagnóstico de Falencias Conceptuales

Leonardo Enrique Castellanos Acuña
Juan Camilo Ramírez Padilla

Trabajo de grado presentado como requisito parcial para optar al título de:
Ingeniero de Sistemas

Directora:
M.Sc. Luz Stella Robles Pedrozo

Línea de Investigación:
Informática educativa
Universidad Tecnológica de Bolívar
Facultad de Ingeniería, Programa de Ing. de Sistemas
Cartagena, Colombia
2017

Agradecimientos

Agradecemos principalmente a Dios por permitirnos culminar con éxito esta etapa de nuestras vidas.

Además queremos expresar nuestra gratitud con las siguientes personas:

A nuestras familias, especialmente a nuestras madres por la excelente educación, sobre todo en valores que nos han brindado, por su apoyo incondicional y por estar siempre a nuestro lado en los momentos más importantes.

A la profesora Luz Stella Robles Pedrozo, M.Sc., director de la tesis, educadora, madre y amiga. Muchísimas gracias por su valioso esfuerzo y dedicación permanente en la realización de este trabajo de grado. **Con usted empezó todo.**

Finalmente, a todos los profesores del programa de Ingeniería de Sistemas, quienes forjaron nuestras capacidades y conocimientos a lo largo de nuestra carrera académica.

Contenido

Agradecimientos	III
Lista de figuras	VII
1 Descripción del Proyecto	2
1.1 Objetivos	2
1.1.1 General	2
1.1.2 Específicos	2
2 Contexto y Motivación	3
2.1 Introducción	3
2.2 Contexto y Descripción del Problema	3
2.3 Marco Teórico	4
2.4 Fundamentos Teóricos de la Implementación	10
2.4.1 Servicios Web	10
2.4.2 Antecedentes	11
2.4.3 REST	13
2.4.4 Principios de diseño	15
2.5 Base Teórica de la Teoría Clásica de Test versus la Teoría de Respuesta al Ítem	18
2.5.1 La Teoría Clásica de Test	18
2.5.2 Debilidades de La Teoría Clásica de Test	19
2.5.3 La Teoría de Respuesta a los Ítems	20
2.6 Estado del Arte	23
2.7 Conclusiones	26
3 Metodología	27
3.1 Introducción	27
3.2 Plan de Trabajo	27
4 Implementación	29
4.1 Introducción	29
4.2 Diseño e Implementación del Motor de Diagnóstico	29
4.2.1 Arquitectura del modelo	29
4.2.2 Modelo de Relación entre Temas y Conceptos	31

4.2.3	Funcionalidad del modelo	34
4.2.4	Implementación del Sistema de Evaluación y Diagnóstico del Aprendizaje	41
4.3	Diseño e Implementación de la Arquitectura	43
4.3.1	Capa de datos	43
4.3.2	Capa de objetos de acceso a los datos (DAO)	43
4.3.3	Capa de negocios	43
4.3.4	REST API	44
4.4	Diseño e Implementación del Servicio Web	44
4.4.1	Recursos/URIs	44
5	Pruebas y Validación	58
5.1	Introducción	58
5.2	Aplicación Web Modelo	58
5.3	Conclusiones	68
6	Conclusiones y Trabajos Futuros	69
6.1	Conclusiones	69
6.2	Trabajo Futuro	70
	Bibliografía	71

Lista de Figuras

2-1. Limitaciones TCL versus Ventajas TRI	21
2-2. Curva Característica del Ítem -CCI-	22
4-1. Arquitectura del Modelo de Evaluación y Diagnóstico	30
4-2. Estructura Jerárquica – Relación entre Conceptos	31
4-3. Relación entre Conceptos – Diagrama Jerárquico Aleatoriedad en Matemáticas	32
4-4. Tabla Relación entre Conceptos - Conceptos C_j afectados por C_i	33
4-5. Tabla Relación entre Conceptos - Conceptos C_i que afectan DIRECTAMEN- TE a C_j	34
4-6. Relación Items-Conceptos	36
4-7. Vector Respuestas Estudiante	37
4-8. Estructura Jerárquica de Conceptos con el valor asignado $PRc(C_j)$	37
4-9. Funcionamiento del Módulo de Diagnóstico	42
4-10. Flujo de Estados DIAGEN	43
4-11. Funcionamiento General del Modelo	56
4-12. Diagrama de Clases DIAGEN	57
4-13. Arquitectura Aplicación Web	57
5-1. Edición del Curso por Instructor	59
5-2. Listado de Cursos por Instructor	59
5-3. Creación de Evaluación por Instructor	61
5-4. Vista Curso de Estudiante	61
5-5. Vista de Materiales de Estudiante	62
5-6. Vista Evaluación de Estudiante	63
5-7. Estructura Jerárquica de la Temática	63
5-8. Tabla Ítem vs Conceptos	64
5-9. Tabla Ítem vs Conceptos Caso 1	64
5-10. PRc Conceptos Caso 1	65
5-11. Diagnóstico Caso 1	65
5-12. Tabla Ítem vs Conceptos Caso 2	65
5-13. PRc Conceptos Caso 2	66
5-14. Diagnóstico Caso 2	66
5-15. Tabla Ítem vs Conceptos Caso 3	66
5-16. PRc Conceptos Caso 3	67

5-17.Diagnóstico Caso 3	67
-----------------------------------	----

1 Descripción del Proyecto

1.1. Objetivos

1.1.1. General

Diseñar e implementar un Sistema Gestor de Aprendizaje (LMS, por sus siglas en inglés) que permitan el montaje de cursos para auto-aprendizaje en estudiantes, e implementa en su módulo de evaluación un sistema de diagnóstico que facilita la identificación de unidades conceptuales en las que presentan falencias los evaluados.

1.1.2. Específicos

1. Establecer un comparativo de LMS, con el fin de identificar particularidades, estándares de diseño, carencias, etc., con el fin de establecer las bases de diseño del repositorio que se propone.
2. Establecer un comparativo de cursos que cumplan con las características del LMS que se propone, con el fin de establecer una metodología de diseño de los tutoriales que pueden formar parte del repositorio que se propone.
3. Seleccionar la herramienta software con la que será implementado el repositorio, teniendo en cuenta facilidades de uso, acceso y estandarización.
4. Diseñar e implementar la arquitectura e interfaz del Sistema y la manera de comunicarse entre sí.
5. Implementar un servicio web (REST) cuyos recursos definirán la funcionalidad del sistema, de acuerdo al diseño propuesto.
6. Presentar una interfaz que consuma el servicio web propuesto para evidenciar su funcionamiento.

2 Contexto y Motivación

2.1. Introducción

Los Sistemas Gestores de Aprendizaje son sistemas, en los que un docente puede elaborar cursos en un entorno que se puede acceder por la web. Los contenidos del curso que se presentan a los estudiantes son relevantes y comprensibles por cada uno de ellos, adicionalmente ofrecen cierta libertad en el momento en que desean estudiar el material.

En la actualidad uno de los medios informáticos utilizados para el desarrollo de habilidades en áreas específicas es la utilización de Sistemas Gestores de Aprendizaje. Por una parte, estos medios han tenido un gran auge e importancia en el campo de la educación, gracias a la creación de algunas plataformas como Codecademy, Microsoft Virtual Academy y Coursera, entre otros; por otra parte, han gestado de manera positiva el aprendizaje de los estudiantes debido a la versatilidad que ofrecen al personalizar el contenido de los cursos (tutoriales) y la flexibilidad en el tiempo de estudio de los mismos.

2.2. Contexto y Descripción del Problema

Es clara la gran utilidad de las plataformas basadas en sistemas gestores de aprendizaje, debido a que han generado un alto índice de aprendizaje en los estudiantes por su accesibilidad y adaptación al estilo de aprendizaje del estudiante con respecto a la forma en la que aprende y del tiempo del que se dispone para estudiar. Sin embargo, estas plataformas no ofrecen a los usuarios la posibilidad de crear sus propios cursos. Además de lo anterior, los métodos de evaluación convencionales usados por este tipo de sistemas no facilitan la detección de los problemas de aprendizaje de los usuarios (estudiantes), lo cual incide en una personalización errónea del material temático y por consiguiente la experiencia del usuario, en el proceso de aprendizaje no será óptima. Por otro lado, en el desarrollo de una asignatura, existen temas de los cuales prescinde el docente, dado que no son parte del contenido base y primordial de la asignatura, pero, sí conforman un complemento para esta. En casos como este, el docente se ve limitado, ya que no puede ni debe hacer uso del tiempo disponible para el desarrollo de la asignatura para explicar estos temas complementarios.

De acuerdo con lo anterior, por medio de este trabajo, se ha propuesto diseñar una prototipo

de un Sistema Gestor de Aprendizaje que implementa en su módulo de evaluación un sistema de diagnóstico que permite identificar las unidades conceptuales en las que presentan falencias los evaluados con el objetivo de: primero, ofrecer una herramienta tecnológica que permita llevar a cabo un proceso de aprendizaje de manera independiente; segundo, ofrecerle al usuario (estudiante) flexibilidad de tiempo en cuanto al desarrollo del curso; tercero, permitir al usuario (estudiante) evidenciar su nivel de avance en el aprendizaje e identificar sus falencias a medida que avanza en el curso (tutorial); cuarto, una vez identificadas las falencias, les da tratamiento sugiriendo al usuario (estudiante) qué conceptos debe trabajar, y quinto, permitir que el usuario (docente) cree de una forma simplificada un curso (tutorial). Esta plataforma (sistema) evaluará generando un diagnóstico personalizado sobre el nivel cognitivo en que se encuentra el estudiante, lo cual permitirá que el contenido estudiante conozca las temáticas que debe reforzar, y en consecuencia el usuario (estudiante) avance a su ritmo de aprendizaje de manera eficiente.

La plataforma permitirá al usuario (docente) crear un curso (tutorial), en primer lugar especificando los temas que lo conforman y cómo se relacionan; y en segundo lugar el docente podrá agregar respectivo el material temático. La plataforma inicialmente permite el registro del estudiante. Por otro lado, la plataforma evaluará al final de cada unidad de un curso la unidad vista. Las preguntas serán de tipo falso o verdadero, ó selección múltiple con única respuesta. Al final de cada examen se le presentará al usuario los conceptos en los que presenta falencias (si los tiene), dándole así la posibilidad de repasar estos conceptos, así como la de ser evaluado nuevamente hasta que logre el nivel de aprendizaje necesario para avanzar en el curso.

2.3. Marco Teórico

Los sistemas de Enseñanza Asistida por Computador (EAC) han sido ampliamente usados para ayudar a los estudiantes a mejorar el rendimiento de su aprendizaje. Los Sistemas de Gestión de Aprendizaje (En inglés Learning Management Systems, LMS) proveen un lugar para enseñar y aprender sin depender del tiempo y el espacio [Sharma and Vatta, 2013] con el fin de proveer a los instructores la posibilidad de diseñar y crear cursos, y a los estudiantes la posibilidad de asistir a ellos en pro de adquirir un determinado conocimiento.

Los requerimientos básicos para el diseño de un LMS son:

Cursos

- Son espacios en donde los instructores se encargan de transmitir conocimientos a un conjunto de estudiantes.
- Los objetos de aprendizaje de un curso deben ser reusables para los usuarios.

Evaluación

- Después de completar el curso los estudiantes deben ser evaluados.
- Además deben tener la posibilidad de evaluar el curso.
- Las respuestas de las evaluaciones deben estar disponibles para el responsable del curso.

Comunicación

- El sistema puede ofrecer chats basados en textos, wikis, foros, entre otros.

Monitoreo del progreso

- El progreso de los estudiantes durante la realización de un curso debe ser rastreado por el sistema.
- Las notas (calificaciones) obtenidas por el estudiante deben estar disponibles tanto para el instructor como para el mismo estudiante.

Administración

- Debe disponer de manejos de roles.
- El acceso al material debe ser responsabilidad de la persona quién mantiene el curso.

Usabilidad

- El sistema debe ser fácil de usar y aprender para los usuarios.
- El sistema debe ser completamente accesible por un navegador web.

Requerimientos técnicos

- El sistema debe poder trabajar bajo una gran carga.
- El sistema debe manejar el crecimiento en el número de usuarios e información.

Los primeros sistemas de Enseñanza Asistida por Computador (EAC) se desarrollaron como una herramienta donde el estudiante pueda leer el material de una materia o tema que desea aprender y como un espacio donde el estudiante pueda practicar los conocimientos adquiridos con una serie de ejercicios propuestos. Luego aparecen los Sistemas Tutores Inteligentes y posteriormente los Sistemas Hipermedia Adaptativos, a continuación se presentan con mayor profundidad:

Sistema Tutor Inteligente (STI)

Un Sistema Tutor Inteligente se concibe como una solución automatizada para la educación. Se buscó emular el comportamiento de un tutor humano, es decir, a través de un sistema que pudiera adaptarse al comportamiento del estudiante, identificando la forma en que él mismo resuelve un problema a fin de poder brindarle ayudas cognitivas cuando lo requiera [Cataldi, Lage, 2009], basándose, para su desarrollo e implementación, en las diferentes técnicas de Inteligencia Artificial.

La mecánica de los STI consiste en guiar al alumno durante su proceso de aprendizaje mediante una estructura que corresponde a un modelo de tutor individual o de ‘uno a uno’ previamente concertado, el cual permite un aprendizaje altamente individualizado y conduce consistentemente a mejores resultados.

Según [Robles Pedrozo and Rodriguez-Artacho, 2012], la arquitectura de un STI, cuenta con los siguientes componentes:

- Modelo del dominio. Corresponde a la respuesta sobre el qué se enseña, y contiene el conocimiento sobre la materia que debe ser aprendida.
- Modelo del alumno. El modelo del alumno proporciona información acerca del estudiante individual, el cual, hace uso de un sistema de aprendizaje basado en computador.

El sistema usa el modelo del estudiante para ayudar a determinar las acciones apropiadas para aquel estudiante. Sin un modelo del alumno, un sistema de aprendizaje basado en computador se ejecutará exactamente de la misma forma o manera con todos los usuarios, debido a que no hay bases para determinar la diferencia que hay entre un usuario y otro. Pero, los estudiantes son diferentes, ellos tienen y vienen con sus propios conocimientos, tienen diferentes intereses, tienen diferentes aptitudes, etc. Un modelo del alumno no tiene que ser completamente exacto para ser útil, ya que una mayor exactitud en el modelo, requeriría de un mayor esfuerzo computacional.

- Modelo de instrucción. Responde a la pregunta ‘Cómo se enseña’. Hace referencia a cómo el sistema debe mostrar el material educativo al alumno.
- Interfaz. A través de ella se lleva a cabo la interacción hombre-máquina.

Hipertexto e hipermedia

La definición del término Hipertexto fue acuñada y aceptada inicialmente como “la presentación de información como una Red de nodos enlazados a través de los cuales los lectores pueden navegar libremente en forma no lineal. Permite la coexistencia de varios autores, desliga las funciones de autor y lector, permite la ampliación de la información en forma casi ilimitada y crea múltiples rutas de lectura” [Nelson, 1972].

Desde otra perspectiva, se entiende por hipertexto como “una tecnología que organiza una base de información en bloques discretos de contenido llamados nodos, conectados a través de una serie de enlaces cuya selección provoca la inmediata recuperación de la información destino” [Díaz et al., 1996].

El hipertexto, entonces, se conforma a partir de métodos tecnológicos para la organización y acceso de la información, de forma que el usuario tenga una gran libertad al momento de

elegir la información que busca entre las abundantes fuentes que se le presentan.

Por su parte, el concepto de hipermedia “resulta ser una combinación de hipertexto y multimedia” [Rouet, 2000]. Para muchos, los términos hipertexto e hipermedia son similares, pero realmente la hipermedia se compone o hace uso del hipertexto en conjunto con información visual, sonora, animación y otras formas de información, desembocando en resultados como los hiperdocumentos.

Sistema Hipermedia Adaptativo (SHA)

Un sistema hipermedia tiene como ancestros a los sistemas hipertexto y multimedia, debido a que integra características de ambos. El hipermedia sería el heredero del hipertexto enriquecido por elementos multimedia [Romero, 2005]. Lo anterior se acoge al hecho de que dicho sistema gestiona y organiza la información en una estructura basada en nodos y enlaces.

“Los nodos son los espacios de información discreta que hay en el hipertexto. El uso del hipertexto parte del supuesto de que textos amplios pueden segmentarse en partes (nodos) de significado propio, de modo que pueden ser leídos e interpretados independientemente. Por su parte, un enlace es la conexión entre dos nodos que proporciona una forma de seguir las referencias entre un origen y un destino” [Romero, 2005].

Un sistema se considera adaptable cuando ofrece ciertas características de adaptación al entorno [Kobsa and Wahlster, 1989]. El hecho de que la visualización de una información en un sistema se pueda modificar en dependencia de alguna característica del usuario, por ejemplo, permite catalogar a dicho sistema como adaptable.

Ahora, un sistema puede ser adaptable, pero no necesariamente adaptativo, y será adaptativo si tiene la capacidad de cambiar a partir de criterios propios, dicho de otra manera, adaptarse de forma automatizada como respuesta a su interacción con un entorno, recibiendo estímulos por parte de este.

Los SHA surgen para personalizar el tratamiento a los usuarios y acomodarse mejor a las necesidades de éstos, respondiendo a casos como, por ejemplo, cuando un usuario se mueve por un espacio de información muy amplio organizado siguiendo las técnicas hipermedia, existe el riesgo de que se desoriente [Ortega Molina, 2005]. Este problema, conocido como la pérdida en el hiperespacio [Nielsen, 1990], es uno de los más frecuentes en la actualidad. Una de las causas que lo originan es la imposibilidad de conocer la organización de la información que contiene el hiperespacio [Oliver, 1995], lo cual conlleva cierta ineficiencia en la navegación. Otra dificultad es la sobrecarga cognitiva [Nielsen, 1998], es decir, las gran cantidad de elecciones que debe hacer el usuario entre los enlaces que encuentra durante su navegación, a cuáles cree que debe acceder inicialmente, a cuales debe acceder después y de cuales debe prescindir.

La forma en que los SHA consiguen adaptarse al usuario, se debe al uso de las técnicas de inteligencia artificial para su implementación. Esto proporciona ventajas como lo es la

orientación del usuario en tiempo y la reducción de las posibilidades de sufrir una pérdida en el hiperespacio.

Los SHA provienen de la integración de los sistemas hipermedia con los Sistemas Tutores Inteligentes (STI), y reúne las ventajas de ambos: el SHA adapta el hiperespacio disponible en función de las características, específicamente, del conocimiento del alumno y, a medida que éste va aprendiendo, la accesibilidad crece, dándole la oportunidad de alcanzar nuevas informaciones [Ortega Molina, 2005].

La Evaluación Mediante Test

Según la definición de la Real Academia de la Lengua Española, un test es una prueba destinada a evaluar conocimientos o aptitudes, en la cual hay que elegir la respuesta correcta entre varias opciones previamente fijadas. Sin embargo, los test -que también pueden ser llamados exámenes o pruebas-, pueden igualmente ser orales o escritos, con preguntas de respuestas abiertas -donde el estudiante responde libremente-, o preguntas de respuestas múltiples -el estudiante debe seleccionar la respuesta correcta de un listado-.

A continuación se hace una descripción de las fases que son recomendables seguir en la construcción de un test de evaluación. Esta descripción va desde la etapa de la redacción de los ítems -como parte fundamental en la concepción del test-, hasta la etapa de la administración del mismo. Posterior a esto, es conveniente realizar periódicamente labores de mantenimiento, en las que la calibración en línea puede llegar ser de mucha utilidad.

- Redacción de preguntas: Para este caso, en el que el docente decide redactar las preguntas del cuestionario que va a aplicar a sus estudiantes, se le recomienda en términos generales, lo siguiente:
 - Evitar el uso de palabras tales como: siempre, generalmente, no, ninguno.
 - Evitar negaciones dobles.
 - Evitar errores gramaticales y ortográficos.
 - Expresar la idea principal de la pregunta en el enunciado.
 - La dificultad de la pregunta no debe estar representada en la comprensión del enunciado.
 - Hacer enunciados precisos, puntuales, evitando colocar información irrelevante.
 - La redacción del enunciado y su pregunta deben conducir a solo una opción como respuesta, y no permitir divergencias.
 - No dejar en el enunciado pistas que sugieran cuál es opción de respuesta correcta.
 - Los enunciados de las preguntas no deben ser excesivamente largos.

- El vocabulario utilizado en los enunciados de las preguntas debe estar al nivel de los evaluados.
- Evitar elementos o gráficos que puedan confundir al evaluado.
- Evitar redacción que pueda herir susceptibilidades con enunciados con sesgo racial, étnico, o de género.

Debe entenderse como opciones distractoras, expresiones u oraciones que son ciertas, pero bajo un contexto que puede ser similar o distinto al que se expresa en el enunciado.

En cuanto a las opciones de respuesta:

- No son recomendables opciones como ‘ninguna de las anteriores’, como tampoco ‘todas las anteriores’.
- De igual forma, se debe evitar hacer uso de más de una opción distractora, así como presentar alguna opción absurda.
- Las opciones deben ser escritas en sentido positivo.
- Se debe evitar que la opción correcta sea la más larga, de esta forma se haría evidente la respuesta acertada.

Como ya se había comentado, las preguntas Tipo I, están compuestas por un enunciado y cuatro opciones de respuesta, designadas con las letras en mayúscula: A, B, C, y D; de las cuales sólo una responde de manera correcta al planteamiento del ítem, o resuelve el problema formulado.

- Aplicación del Examen - Test: Una vez redactados los ítems o preguntas del examen, la aplicación de éste puede hacerse de la manera tradicional, o digital haciendo uso del computador. Interesa para el presente trabajo, aplicar el examen de manera tradicional, la cual consiste en entregar en físico, -papel-, la hoja u hojas del examen, que bien puede ser en forma de cuadernillo dependiendo de la extensión de los temas a evaluar, a los estudiantes.

La razón por la cual el examen se aplica en físico, es porque la población a la que podría ir dirigida esta investigación, es a instituciones educativas que cuentan con recursos informáticos insuficientes como para garantizar: conexión en línea 24 horas, suficiencia de equipos de computadoras, disponibilidad de equipos de computadoras, competencias mínimas en el manejo de herramientas tecnológicas; y en esta misma situación se encuentran una gran mayoría de instituciones educativas en el país.

Sin embargo, frente a esta limitante, el proceso de retroalimentación a los estudiantes, para identificar falencias en el proceso de aprendizaje de ciertos temas, sigue siendo posible, así

el método de aplicación del examen sea el tradicional. De todas formas, está latente la necesidad en los docentes de las instituciones educativas, el que luego de aplicar un examen, se pueda retroalimentar a sus estudiantes de manera específica a cada uno, dependiendo del nivel de conocimiento o desconocimiento que tengan de manera individual cada estudiante, o expresado de otra forma, este proceso puede ser llamado Retro-alimentación Inteligente.

2.4. Fundamentos Teóricos de la Implementación

2.4.1. Servicios Web

Un Servicio Web es un componente software que puede ser registrado, descubierto e invocado mediante protocolos estándares de Internet. Permiten exponer y hacer disponibles funcionalidades (servicios) de los sistemas informáticos de las organizaciones mediante tecnologías y protocolos WEB estándar. Cada Servicio Web se responsabiliza de realizar un conjunto de funciones concretas y bien definidas. Los Servicios Web actúan como componentes independientes que se pueden integrar para formar sistemas distribuidos complejos.

El World Wide Web Consortium (W3C) define a un Servicio Web cómo “Una aplicación software identificada por un URI (Uniform Resource Identifier), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los Servicios Web hacen posible la interacción entre agentes software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet.”

Un Servicio Web presenta las siguientes características:

- **Interoperabilidad:** Proporcionan funcionalidades e intercambio de datos entre distintas aplicaciones, en lenguajes de programación diferentes, ejecutadas sobre cualquier plataforma. No imponen restricciones sobre las aplicaciones a las que dan acceso ni sobre las tecnologías que las implementan (independencia de lenguaje y de plataforma).
- **Uso de estándares abiertos:** Un Servicio Web se asienta sobre protocolos y estándares ya existentes y muy difundidos (HTTP, XML, JSON, por ejemplo). OASIS y W3C: organizaciones responsables de definir la arquitectura y estándares para los Servicios Web.
- **Mínimo acoplamiento:** Un Servicio Web debe procurar que, ante posibles modificaciones en su diseño, los sistemas consumidores se vean impactados lo mínimo posible. Esto se logra a través del encapsulamiento de una o más aplicaciones ofreciendo un interfaz único accesible por la Web; También, ofreciendo una interfaz pública y estable, independiente de su implementación concreta.

2.4.2. Antecedentes

En noviembre del año 1990, Sir Tim Berners-Lee, actual director del World Wide Web Consortium, recibiendo ayuda por parte de Robert Cailliau, publicó una propuesta formal en la cual presentaba lo que ahora conocemos como World Wide Web. Desde entonces, la World Wide Web ha estado usando HTTP.

Hyper Transfer Protocol (HTTP), según la IETF [Fielding, 1997], es un protocolo, a nivel de aplicación, para sistemas de información hipermedia, distribuidos y colaborativos. Es un protocolo genérico, sin estado y orientado a objetos, el cual puede ser usado para múltiples tareas a través de la extensión de sus métodos de petición. Una característica de HTTP es el arreglo y negociación de la representación de la información, lo cual permite que la implementación de un sistema sea independiente de los datos que se manejen y transfieran.

Berners-Lee [Berners-Lee, 1996] fue uno de los autores de la especificación estándar para HTTP/1.0 y HTTP/1.1 [Fielding, 1997]. Otro autor fue Roy Thomas Fielding quien, al mismo tiempo que se desarrollaba la especificación estándar RFC 2068 [Fielding, 1997], perfeccionó su propio modelo de arquitectura de software, presentando una serie de principios, propiedades y restricciones [Fielding, 2004]. Con lo anterior, nació el término REpresentational State Transfer (REST), que se introdujo en el año 2000 en su tesis doctoral [Fielding, 2000].

Teniendo en cuenta la relación entre REST y HTTP, y que la World Wide Web usa HTTP, REST puede considerarse como el conjunto de principios base para la Web. Muchos de los Servicios Web de los últimos años han olvidado o ignoran la característica fundamental que hace a la Web exitosa: su simplicidad. Estos servicios se basan en arquitecturas complejas y pesadas para el acceso a objetos distribuidos, similares a COM o CORBA [Richardson and Ruby, 2007]. REST trata de ser fiel a los principios de la Web.

Otros estilos de arquitectura de software para Servicios Web son:

Remote Procedure Calls (RPC, Llamadas a Procedimientos Remotos). Un RPC [Ramaratnam, 2007] es un protocolo que permite a un programa “llamar” a un módulo de software ubicado de forma local (el módulo se halla en la misma máquina) o en externa (el módulo se halla en otra máquina). Fue descrito por primera vez en el año 1976, en especificación estándar RFC 707. Las primeras implementaciones de RPC fueron realizadas por Xerox en el año 1991, dando como resultado su protocolo Courier; también hubo una implementación en UNIX, realizada sobre el sistema de redes computacionales (Network Computing System, NCS) de Apollo Computer Inc.; ésta última se convirtió en la base para el desarrollo del Modelo de Computación Distribuida, propuesto por la Open Software Foundation (OSF). Los Servicios Web basados en RPC presentan una interfaz de llamada a procedimientos y funciones distribuidas. Típicamente, la unidad básica de este tipo de servicios es la operación WSDL. Las primeras herramientas para Servicios Web estaban

centradas en esta visión. Sin embargo, fue criticado por no ser débilmente acoplado, ya que suele ser implementado por medio del mapeo de servicios directamente a funciones específicas del lenguaje o llamadas a métodos.

Arquitectura Orientada a Servicios (Service-oriented Architecture, SOA). SOA se define [Bianco et al., 2007] como un estilo de arquitectura donde el componente básico del sistema es el usuario del servicio y proveedor de servicio. Al contrario que los Servicios Web basados en RPC, este estilo es débilmente acoplado, debido a que se centra en el “contrato” proporcionado por el documento WSDL, más que en los detalles de implementación subyacentes. SOA impone las siguientes restricciones:

- Los usuarios del servicio envían peticiones a los proveedores de servicio. Un proveedor de servicio puede ser un usuario del servicio.
- Un usuario del servicio puede descubrir dinámicamente nuevos proveedores de servicios a través de un directorio de servicios.
- Un ESB (Enterprise Service Bus) se encarga de supervisar la interacción entre los usuarios y proveedores.

El W3C, mediante el documento WSA (Web Services Architecture) [9] define a un Servicio Web como un sistema de software diseñado para soportar interacciones interoperables de máquina a máquina a través de una red. Este sistema tiene una interfaz descrita en un formato procesable por la máquina (específicamente WSDL). Otros sistemas interactúan con el Servicio Web usando mensajes SOAP, usualmente transportados usando HTTP con una serialización XML, en conjunto con otros estándares relacionados con la Web.

El documento WSA no especifica cómo implementar un Servicio Web, y tampoco establece ninguna restricción con respecto a cómo pueden combinarse los Servicios Web. Solo se describen las características mínimas que los Servicios Web tienen en común y también, aquellas características que son requeridas por algunos de ellos. En cambio, la Web Services Interoperability Organization (WS-I) en su definición de Servicio Web, impone el uso de SOAP y WSDL.

De acuerdo con el W3C, se tienen dos clases de Servicios Web: Servicios Web obedientes a REST, y Servicios Web arbitrarios [Richardson and Ruby, 2007]. Ambas clases usan URIs (para identificar recursos), protocolos web (como HTTP y SOAP 1.2), y formatos XML para mensajería.

Para expandir las capacidades de un Servicio Web se tienen un número de estándares, protocolos y especificaciones, en su mayoría creadas sobre HTTP. Todo lo anterior se conoce como la “WS Protocol Stack” o pila de protocolos de Servicios Web, la cual incluye protocolos como WS-Notification, WS-Security, WSDL y SOAP [Richardson and Ruby, 2007]. Esta

pila de protocolos ha evolucionado rápidamente, agregando complejidad a la simplicidad característica de la Web.

SOAP, inicialmente entendido como Simple Object Access Protocol, ahora es solo un acrónimo que, según [Gudgin, 2007], es un protocolo ligero cuyo propósito es el intercambio de información estructurada, en un ambiente distribuido y descentralizado. SOAP hace uso de XML (Extensible Markup Language) para definir un framework de mensajería que provee una estructura para la construcción de mensajes, los cuales serán intercambiados a través de múltiples protocolos.

Los mensajes SOAP son usados como el empaque donde la aplicación anexa cualquier información que requiera ser enviada [Alonso et al., 2004]. El empaque contiene una cabecera (header) y un cuerpo (body). Se usa un esquema XML para describir la estructura del mensaje SOAP, a fin de que los sistemas que estén comunicándose, puedan ensamblar y desensamblar el contenido del mensaje y darle el tratamiento correspondiente.

WSDL (Web Services Description Language), en su versión 2.0 provee un modelo y un formato XML para la descripción de Servicios Web [Christensen, 2001]. WSDL 2.0 hace posible la separación entre la descripción de la funcionalidad abstracta ofrecida por el servicio, y los detalles específicos de la descripción del servicio (“cómo” y “dónde” se ofrecen las funcionalidades).

2.4.3. REST

REST es el acrónimo de Transferencia de Estado Representacional (REpresentational State Transfer), término usado por Roy Fielding (uno de los creadores de HTTP) en su tesis doctoral para describir un estilo de arquitectura que utilizar como modelo en los sistemas de computación Web. No es un estándar, sino un enfoque que muestra cómo desarrollar y proporcionar servicios en Internet, por tanto considerado como un estilo arquitectónico para diseño de software a gran escala. El propio autor lo define de manera concisa en éste párrafo [Fielding, 2000]:

“REST es un intento de mostrar cómo debe comportarse una aplicación Web bien diseñada: una red de páginas Web (una máquina de estados virtual) donde el usuario progresará seleccionando enlaces (transiciones de estado) que devuelven la página siguiente (el siguiente estado de la máquina) que el usuario manejará a su gusto”.

En su tesis, el autor hace énfasis en las características concretas que han hecho exitoso el uso de los cuales coinciden con el enfoque REST y que son, según su criterio:

- **Escalabilidad de la interacción con los componentes:** La Web ha crecido exponencialmente sin degradar su rendimiento. Una prueba de ellos es la variedad de clientes que pueden acceder a través de la Web: estaciones de trabajo, sistemas industriales, dispositivos móviles, etc.
- **Generalidad de interfaces:** Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial. Esto no es del todo cierto para otras alternativas, como SOAP para los Servicios Web. Puesta en funcionamiento independiente: Este hecho es una realidad que debe tratarse cuando se trabaja en Internet. Los clientes y servidores pueden ser puestas en funcionamiento durante años. Por tanto, los servidores antiguos deben ser capaces de entenderse con clientes actuales y viceversa. Diseñar un protocolo que permita este tipo de características resulta muy complicado. HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs, a través de la habilidad para crear nuevos métodos y tipos de contenido.
- **Puesta en funcionamiento independiente:** Este hecho es una realidad que debe tratarse cuando se trabaja en Internet. Los clientes y servidores pueden ser puestas en funcionamiento durante años. Por tanto, los servidores antiguos deben ser capaces de entenderse con clientes actuales y viceversa. Diseñar un protocolo que permita este tipo de características resulta muy complicado. HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs, a través de la habilidad para crear nuevos métodos y tipos de contenido.
- **Compatibilidad con componentes intermedios:** Los más populares intermediarios son varios tipos de proxys para Web. Algunos de ellos, las caches, se utilizan para mejorar el rendimiento. Otros permiten reforzar las políticas de seguridad: firewalls. Y por último, otro tipo importante de intermediarios, gateway, permiten encapsular sistemas no propiamente Web. Por tanto, la compatibilidad con intermediarios nos permite reducir la latencia de interacción, reforzar la seguridad y encapsular otros sistemas.

Para satisfacer los objetivos anteriores, existe una serie de normas a seguir, los cuales son:

- **Identificación de recursos y manipulación de ellos a través de representaciones:** Esto se consigue mediante el uso de URIs. HTTP es un protocolo centrado en URIs. Los recursos son los objetos lógicos a los que se le envían mensajes. Los recursos no pueden ser directamente accedidos o modificados. Más bien se trabaja con representaciones de ellos. Cuando se utiliza un método PUT para enviar información, se coge como una representación de lo que nos gustaría que el estado del recurso fuera. Internamente el estado del recurso puede ser cualquier cosa desde una base de datos relacional a un fichero de texto.
- **Mensajes autodescriptivos:** REST dicta que los mensajes HTTP deberían ser tan descriptivos como sea posible. Esto hace posible que los intermediarios interpreten los

mensajes y ejecuten servicios en nombre del usuario. Uno de los modos que HTTP logra esto es por medio del uso de varios métodos estándares, muchos encabezamientos y un mecanismo de direccionamiento. Por ejemplo, las cachés Web saben que por defecto el comando GET es cacheable (ya que es idempotente) en cambio POST no lo es. Además saben cómo consultar las cabeceras para controlar la caducidad de la información. HTTP es un protocolo sin estado y cuando se utiliza adecuadamente, es posible interpretar cada mensaje sin ningún conocimiento de los mensajes precedentes. Por ejemplo, en vez de loguearse del modo que lo hace el protocolo FTP, HTTP envía esta información en cada mensaje.

- **Hipermedia como un mecanismo del estado de la aplicación:** El estado actual de una aplicación Web debería ser capturada en uno o más documentos de hipertexto, residiendo tanto en el cliente como en el servidor. El servidor conoce sobre el estado de sus recursos, aunque no intenta seguirle la pista a las sesiones individuales de los clientes. Esta es la misión del navegador, el cuál sabe cómo desplazarse de recurso a recurso, recogiendo información que el necesita o cambiar el estado que el necesita cambiar.

2.4.4. Principios de diseño

Para la implementación de un Servicio Web bajo la arquitectura REST, se tienen los siguientes principios de diseño [Richardson and Ruby, 2007, Pautasso et al., 2008]:

Recursos

Todo lo que un servicio provee es un recurso. Un recurso puede ser un objeto físico o un concepto abstracto. Todo recurso tendrá al menos un URI (un identificador único). Si una pieza de información no tiene una URI, entonces no será un recurso, y no estará en la web, dado que no puede accederse a él. Dos recursos no pueden tener el mismo URI (Troyano fig 2.1), pero dos URIs diferentes pueden apuntar a la misma información al mismo tiempo (Troyano fig 2.2). El URI debería tener una estructura, a fin de que el cliente pueda predecir el comportamiento del recurso y sea capaz de realizar peticiones al servicio.

Un recurso puede representarse en cualquier formato, cualquier tipo de multimedia, como por ejemplo: XML, XHTML, JSON o CSV. Cada representación tendrá su propia dirección pero sin indicar el formato, por ejemplo: La información de las URIs `dir/to/resource/res.xml` y `dir/to/resource/res.json` puede ser recuperada bajo una misma URI, `dir/to/resource/res`, dado que en la petición, tanto en el envío como en la respuesta, se especifican cabeceras (headers) que brindan detalles sobre la información que se requiere/se recibe. En este caso, la cabecera “Content-Type” indica al cliente el tipo de representación de la información. Así, el cliente sabrá que mecanismo debe usar para “lee” la información recibida correctamente.

Direccionabilidad

Todo recurso debería ser direccionable. Un recurso es direccionable a través de URIs, por lo que tendrán tantas URIs como tantos recursos haya.

Sin estado

Toda solicitud HTTP debe suceder completamente aislada. Lo anterior implica que, al realizarse una solicitud HTTP, toda la información requerida para procesar dicha solicitud debe estar disponible, es decir, la solicitud NO dependerá de la ejecución exitosa de una solicitud previa. Dado el caso en que se requiera información obtenida a través de una solicitud previa, es necesario realizar dicha solicitud nuevamente, obtener la información, y enviarla en una solicitud posterior. La mecánica anterior hace que un sistema sea simple, confiable y horizontalmente escalable.

Por ejemplo, un cliente realiza una solicitud a un servidor A. El cliente realiza otra solicitud al servidor A, pero el servidor falla al procesar la solicitud. En este caso, un servidor B podrá encargarse de atender la solicitud debido a que toda la información necesaria para procesarla proviene siempre del cliente. Dado que el servidor no almacena información del cliente, cualquier servidor puede procesar cualquier solicitud, facilitando la adición de nuevos servidores. Esto permite que el servidor sea reemplazable, lo que hace al sistema escalable.

Conectividad

Un servicio RESTful debe guiar al cliente de un estado a otro, enviando enlaces en las respuestas (representación) de cada solicitud. Una representación consiste en un documento hipertexto que, además de contener la información solicitada por el cliente, debe proveer enlaces a otros recursos.

Todos los recursos, identificador por un URIs, deberían estar conectados y enlazados, con el objetivo de que el cliente pueda descubrir la interfaz del sistema accediendo a los recursos provistos en la respuesta de cada solicitud.

Interfaz uniforme

El cliente interactúa con el servidor a través de los siguientes métodos HTTP: GET, POST, PUT, DELETE. Estos métodos definen las operaciones posibles para un recurso.

1. **HTTP GET.** Este método devuelve cualquier información que sea identificada a través de una URI [Fielding, 1999]. Esto es, la URI identifica a un recurso específico. Por ejemplo, la siguiente URI “publicaciones/1412” permite identificar a una publicación con un identificador o código (1412). En el caso de los servicios RESTful, la información identificada por una URI es una representación de un recurso. El método GET es seguro ya que no altera el estado del servidor y solo recupera información; y es idempotente debido a

que, sin importar cuantas veces se ejecute una solicitud (enviando la misma información) a través de este método, siempre se obtendrá la misma respuesta que se conseguiría al realizarla una sola vez.

Dado que el protocolo HTTP es “sin estado”, el término seguro es semejante a idempotente. Ser idempotente permite repetir solicitudes de tipo GET de forma segura. Si una solicitud GET es exitosa, se obtendrá, además de la información solicitada, un estado de respuesta “200” (OK). Si el recurso solicitado no existe, el estado de respuesta será 404 (No encontrado).

2. **HTTP POST.** El método POST se usado para solicitar la creación de un nuevo recurso, teniendo en cuenta la información enviada en la solicitud. Por lo tanto, la información enviada se usará para identificar al nuevo recurso en solicitudes tipo GET, PUT y DELETE.

La ejecución de una solicitud POST es exitosa si el servidor pudo crear el nuevo recurso, y en este caso, el servidor retornará como respuesta, una representación del recurso creado (datos del recurso y una URI de referencia), junto con el código de respuesta “201” (Creado).

3. **HTTP PUT.** El método PUT solicita que la información enviada en la solicitud, sea almacenada en el recurso identificado por la URI. Si la URI usada en la solicitud identifica a un recurso existente, la información enviada debe considerarse como una versión modificada del recurso que se encuentra en el servidor. En el caso contrario, si la URI no apunta a un recurso existente, el método PUT puede crear un nuevo recurso con la información enviada (esta capacidad es configurable).

Aparentemente los métodos POST y PUT realizan el mismo procedimiento, pero se diferencian en función de su significado. La URI en una solicitud de tipo PUT identifica a un recurso en función de la información enviada por el cliente (el cliente debe indicar, por ejemplo, un código único que permita diferenciar al recurso). En contraste, la URI en una solicitud de tipo POST identifica al recurso que procesará la información enviada por el cliente. Por otra parte, el método PUT es idempotente, dado que al ejecutar la solicitud muchas veces, se obtendrá el mismo resultado conseguido en la primera ejecución.

4. **HTTP DELETE.** El método DELETE permite solicitar al servidor la eliminación de un recurso identificado por la solicitud. Este método es idempotente al igual que los métodos PUT y GET. Si la solicitud es exitosa, se obtendrá como respuesta un estado 200 (OK). Si la solicitud falla, la respuesta debería contener un mensaje de error explicando la naturaleza del problema.

2.5. Base Teórica de la Teoría Clásica de Test versus la Teoría de Respuesta al Ítem

2.5.1. La Teoría Clásica de Test

Esta teoría tiene sus inicios a comienzos del siglo XX, con Spearman en [1904, 1907, 1913]. Spearman propone un modelo simple para las puntuaciones que obtienen las personas luego de aplicar un test, éste consiste en asumir que la puntuación que obtiene una persona en un test -llamada puntuación empírica y representada con una 'X'-, está formada por dos componentes: una puntuación que obtiene la persona luego de aplicar a un test, representada por la letra 'V' , y un error, representado por la letra 'e', este error puede deberse a muchas causas que en la mayoría de los casos no se puede controlar. Formalmente, la puntuación empírica puede expresarse como:

$$X = V + e$$

Sin embargo, cuando una persona presenta un test y obtiene, por ejemplo 80 puntos, en este punto lo único que se obtiene es la puntuación empírica. Así como está planteado el modelo, éste no permite saber ni cuál es su puntuación verdadera, ni cuál es el error cometido. Este error puede deberse a muchas razones que inclusive pueden estar en el contexto, en el test o en la propia persona.

Para lo anterior, Spearman añade tres supuestos al modelo. El primer supuesto consiste en definir la puntuación verdadera como la esperanza matemática de la puntuación empírica. Spearman define la puntuación verdadera -de una persona en un test-, como aquella puntuación que obtendría como medida si se le aplicara infinitas veces el test. Se expresa así:

$$V = E(X)$$

Para el segundo supuesto, Spearman asume que no existe relación entre el cálculo de las puntuaciones verdaderas de las personas y el tamaño de los errores que afectan esas puntuaciones. Esto mismo puede expresarse de varias formas: la correlación entre el puntaje verdadero y el puntaje de error, es cero; o, el valor de la puntuación verdadera de una persona no tiene nada que ver con el error que afecta esa puntuación; o lo que es lo mismo, puede haber puntuaciones verdaderas altas con errores bajos, o puntuaciones verdaderas altas con errores altos. Puede expresarse así:

$$r(v, e) = 0$$

Para el tercer supuesto, Spearman establece que los errores de medida de las personas en un test no están relacionados con los errores de medida en otro test distinto. Se expresa como:

$$R(e_j, e_k) = 0$$

Adicional al modelo y los tres supuestos, se formula la definición de los Test Paralelos, que son aquellos tests que miden lo mismo exactamente, pero con ítems distintos. Esta definición

sugiere que las puntuaciones verdaderas de las personas obtenidas de tests paralelos, serían las mismas, como también serían iguales las varianzas de los errores de medida.

La esencia de la Teoría Clásica de los Tests la constituyen los planteamientos hechos por Spearman, explicados en párrafos anteriores: el modelo lineal, junto con los tres supuestos, y la definición de tests paralelos.

2.5.2. Debilidades de La Teoría Clásica de Test

Nuevas teorías surgen a pesar de la utilidad y eficacia de la aplicación de la TCT en casi 100 años. Estos nuevos enfoques se dan más para complementar la TCT, que para contradecirla, porque habían dos cuestiones básicas que no encontraban buena solución en la teoría clásica y que hacían que la medición -en el campo de la psicología-, no fuera homologable a la medición que exhibían otras ciencias empíricas [Muñiz Fernández, 2010], son ellas:

- Las mediciones no resultan invariantes respecto al instrumento utilizado, en otras palabras, las mediciones resultan variantes respecto al instrumento, o tal como Muñiz lo explica en [Muñiz Fernández, 2010], si se evalúa la inteligencia de tres personas distintas con un test diferente para cada persona, los resultados no son comparables, no se puede decir qué persona es más inteligente porque no se tiene la misma base de medida. Lo más deseable científicamente sería que los resultados obtenidos al utilizar distintos instrumentos estuvieran en la misma escala.
- La ausencia de invarianza de las propiedades de los test respecto de las personas utilizadas para estimarlas; es decir, los valores de las estadísticas de los ítems y del test dependen de la muestra de examinados; o como lo explica [Muñiz Fernández, 2010], propiedades psicométricas importantes de los test, tales como la dificultad de los ítems o la fiabilidad del tests, están en función del tipo de personas utilizadas para calcularlas. Por ejemplo, la dificultad de los ítems, o los coeficientes de fiabilidad dependen en gran medida del tipo de muestra utilizada para calcularlos.

Estos aspectos o dificultades son solventados por la TRI. Adicionalmente, otras situaciones problemáticas también se presentan, por ejemplo, en la TCT cuando se calcula el coeficiente de fiabilidad de un test, tal como el alfa de Cronbach [Cronbach, 1951], se está sugiriendo que ese test está midiendo con una cierta fiabilidad a todas las personas evaluadas con el test, cuando, [Muñiz Fernández, 2010] se tiene evidencia empírica suficiente que indican que los tests no miden con la misma precisión a todas las personas; y más bien la precisión viene a depender en gran medida del nivel que muestre la persona en la variable medida. Esta situación, nuevamente es solucionada por la TRI, a través de ‘La Función de Información’, que permite estimar la fiabilidad del test aplicado en función del nivel de la persona en la variable que se mide. Más adelante se explicará en detalle las particularidades de esta función, y cómo puede ser utilizada.

Seguidamente, se hará una introducción a la Teoría de Respuesta al Ítem, desde sus inicios, su gran utilidad, y principalmente cómo complementa los espacios no concretados por la TCT. Finalmente, y teniendo en cuenta que esta teoría -la TRI-, es base fundamental en el desarrollo de la presente investigación, temas relacionados con los modelos que propone la teoría, sus supuestos, y su aplicabilidad, serán presentados en secciones apartes, más adelante.

2.5.3. La Teoría de Respuesta a los Ítems

La Teoría de Respuesta al Ítem -TRI-, tuvo sus primeros estudios con trabajos pioneros de Thurstone, en los años 20 [Thurstone, 1925]. Pero, fue ya en los años cincuenta con el psicómetra Frederic Lord, en [Lord, 1952], quien -a través su tesis doctoral-, haga más formal los estudios y postulados de la TRI. En 1968, Lord y Novick le dan un gran impulso a la teoría en [Lord and Novick, 1968], y a partir de ahí comienzan a aparecer programas informáticos creados para encargarse de los cálculos que implica hacer uso de los modelos de la TRI, son ellos: BICAL y LOGIST en 1976, BILOG en 1984, MULTILOG en 1983, entre otros.

Las razones que explican el que la TRI ha tomado tanto tiempo en posicionarse, a pesar de tener sus orígenes tan cercanos a la TCT, son, como señala [Navas, 1994], por un lado, a que la TRI no se desarrolló en un contexto vinculado a las teorías de la inteligencia -como es el caso de la TCT-, sino vinculada a problemas técnicos en la construcción de test y en la estadística matemática [Embreston, 1985], y por otro lado, a su complejidad matemática, que no la hacía llegar fácilmente a los lectores no iniciados en el tema; sin embargo, lo que más afectó su avance, se debió a que el soporte matemático, informático y tecnológico que necesitaba esta teoría para ser implementada, hizo que careciera en su momento de procedimientos prácticos necesarios para ser utilizada.

Esta teoría surge como una reacción a los problemas y limitaciones que presentaba la Teoría Clásica de los Tests. El objetivo principal de la TRI es conseguir medidas invariantes respecto de los sujetos medidos y de los instrumentos utilizados [Muñiz Fernández, 1997]. [Robles Pedrozo and Rodriguez-Artacho, 2012] presentan un cuadro comparativo que contrasta las limitaciones de la TCT versus los planteamientos de la TRI para superarlos. Ver Figura 2-1.

Por las ventajas que la TRI presenta frente a la TCT, es la teoría hegemónica en la investigación en psicometría [Santisteban and Alvarado, 2001], y como Muñiz complementa en [Muñiz Fernández, 1997], diversos modelos de la TRI se han utilizado con mucho éxito en el diseño de test de rendimiento educativo de aplicación recurrente, en la generación de test adaptativos -TAIs-, en la investigación sobre funcionamiento diferencial de los ítems -DIF-, y en la construcción de bancos de ítems.

Nro.	Limitaciones TCT	Explicación desde la perspectiva de la TCT	Ventajas TRI
1	Las mediciones no resultan invariantes respecto al instrumento utilizado.	La puntuación que un evaluado obtiene <u>depende del conjunto de ítems</u> que le fueron aplicados. Por ejemplo: la puntuación que obtiene una persona, a la que se le aplican dos test, que aunque midan la misma característica, al final tales puntuaciones van a ser diferentes si los ítems que conforman cada test, tienen niveles de dificultad diferentes. Esta particularidad hace muy difícil comparar estas puntuaciones, porque éstas se encuentran amarradas y sólo podrán interpretarse en relación al test del que fueron obtenidas.	El nivel de habilidad de una persona puede ser obtenido a partir de conjuntos de ítems distintos.
2	La ausencia de invarianza de las propiedades de los test respecto de las personas utilizadas para estimarlas.	Las características de los ítems <u>van a depender del grupo de personas</u> a quienes se les apliquen los ítems. Por ejemplo, si se desea calcular el índice de dificultad de un ítem que mide el conocimiento en gramática del español, el índice calculado a un grupo de estudiantes de Licenciatura en Castellano, va a ser diferente al índice calculado en un grupo de estudiantes de medicina.	Independientemente de cuál sea la distribución de los niveles de rasgo, se obtendrán las mismas estimaciones de los parámetros de los ítems. Los parámetros de los ítems deben ser los mismos tanto si éstos son aplicados a un grupo de personas con alto nivel de rasgo, o a un grupo con bajo nivel de rasgo.
3	La TCT supone el error de medida como una propiedad del test, y por tanto, ésta es igual para todos los sujetos.	La TCT asume que el error es el mismo para todos los sujetos. Como cuando se calcula un coeficiente de fiabilidad de un test en la TCT, asumiendo que el test está midiendo con la misma fiabilidad a todas las personas. Los tests no miden con la misma precisión a todos los evaluados.	La TRI permite obtener la precisión con la que cada persona es medida. La TRI presenta la Función de Información, que sustituye al coeficiente de fiabilidad de la TCT.

Figura 2-1: Limitaciones TCL versus Ventajas TRI

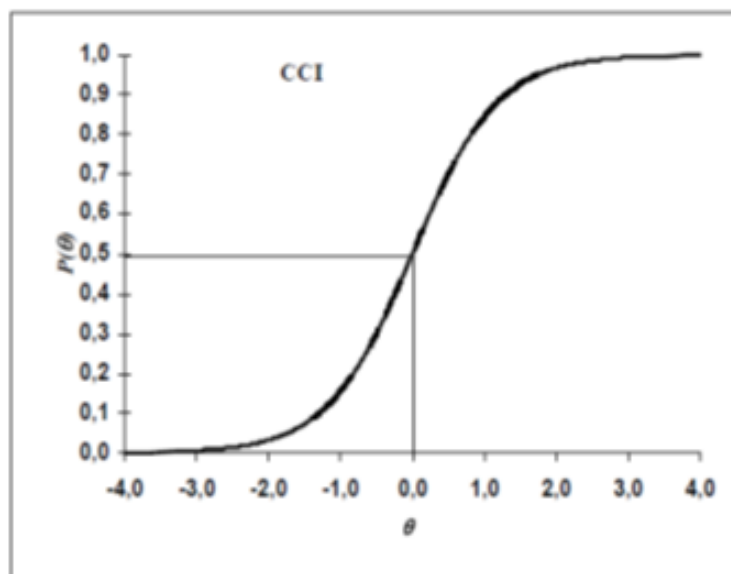


Figura 2-2: Curva Característica del Ítem -CCI-

La TRI se apoya en dos principios fundamentales [Hambleton et al., 1991], son éstos:

- Los resultados obtenidos por un individuo en un test pueden ser explicados mediante un conjunto de factores denominados rasgos latentes o habilidades.
- La existencia de una relación funcional entre los valores de la variable que miden los ítems -resultados del alumno en el test-, y la probabilidad de acertar éstos. Esta relación puede ser descrita mediante una función monótona creciente denominada Curva Característica del Ítem -CCI-.

La CCI es la función básica de la TRI, y representa la probabilidad condicional de que un evaluado, con cierto nivel de rasgo latente estimado -representado con la letra griega θ -, responda correctamente a un ítem. Esta probabilidad depende, evidentemente, de cuál sea el nivel de la persona en la variable medida. Un ejemplo que represente la relación funcional, puede verse en la Figura 2-2, que muestra el comportamiento de la curva característica. En esta gráfica puede verse que al aumentar los valores de la variable medida -denominada θ -, aumenta la probabilidad de acertar el ítem $P(\alpha)$. Los valores de la variable medida se encuentran en el intervalo de $(-\infty, \infty)$, mientras que en la TCT los valores dependían de la escala de cada test -desde un el valor mínimo obtenible, hasta un máximo-.

La forma concreta o particular de la Curva Característica del Ítem se determina por el valor que tomen tres parámetros o características de cada ítem -parámetros de los que más adelante se hablará-. Bajo esta teoría, se tiene que:

- Las características del ítem son independientes de la distribución de la aptitud en la población de sujetos.

- La medida del evaluado no depende de los parámetros de los ítems, ni de la medida de los demás sujetos.
- La curva característica de cada ítem es independiente del objeto medido.

Esta independencia entre el instrumento de medición y los sujetos, es la diferencia esencial entre las teorías TRI y TCT.

2.6. Estado del Arte

El desarrollo de internet ha influenciado sustancialmente en el diseño de entornos e interfaces en los Sistemas Tutores Inteligentes. En efecto se han visto multiplicadas con la aparición de las capacidades hipermedia y multimedia [Brusilovsky, 1999]. En la década de los 90 se iniciaron investigaciones en el campo de los Sistemas Hipermedia Adaptativos (SHA) y esto se debió precisamente al uso generalizado de internet [Brusilovsky and Weber, 2001].

Estos han tenido, por una parte, un gran auge e importancia en el campo de la educación. En la actualidad existen múltiples Plataformas- Repositorios de Tutoriales que tienen como fin de facilitar el aprendizaje de ciertos temas, a través de la Web. A continuación se describen algunas de estas plataformas.

1. REPOSITORIO OBJETOS DE APRENDIZAJE: Repositorio Objetos de Aprendizaje [Repositorio de Objetos de Aprendizaje, 2016] es una plataforma desarrollada en el año 2007 por el Instituto Tecnológico de Sonora (ITSON), como parte del Fideicomiso SEP – UNAM (Secretaría de Educación Pública – Universidad Nacional Autónoma de México), con la finalidad de crear entre distintas universidades un repositorio de objetos de aprendizaje (cursos) con carácter nacional. Cada tutorial plantea las capacidades y conocimientos que se espera que el usuario alcance al finalizarlo. Al iniciar el curso se realiza un evaluación diagnostica para medir el nivel de conocimiento del usuario previo a la adquisición de los conceptos de la temática; la evaluación consiste en un conjunto de preguntas de selección múltiple (Inmediatamente se responde la pregunta, se evalúa). Posee a su vez varias secciones donde se explica de forma detallada cada concepto, no existe limite de tiempo para su uso y se puede acceder a ellas en cualquier orden.
2. MOODLE: Moodle [Moodle, 2016] Es un paquete de software diseñado para ayudar a los educadores a crear cursos y plataformas educativas en línea. Permite la creación masiva de cursos. En estos se pueden agregar material educativo en diferentes formatos (Videos, imágenes, documentos, etc). Los cursos no tienen un estandar establecido, es flexible en cuánto su creación, desarrollo y evaluación. Ofrece la posibilidad de crear evaluaciones de diversos estilos, preguntas con múltiples opciones y única respuesta, preguntas con múltiples opciones y multiples respuestas, autocompletar. Las restricciones como: tiempo

de ejecución, número de intentos disponibles, etc; también son definidos por el profesor. Permite a los aprendices interactuar por medio de foros y chats. Los educadores y los educandos pueden monitorear el progreso y el grado de finalización con un conjunto de opciones para monitoreo de actividades individuales o recursos, y también a nivel del curso. Ofrece una interfaz diseñada para ser responsiva y accesible, es fácil de navegar, tanto en computadoras de escritorio como en dispositivos móviles.

3. MICROSOFT VIRTUAL ACADEMY (MVA): MVA [MVA, 2016] es un sistema de aprendizaje proporcionado por Microsoft que proporciona capacitación gratuita a los desarrolladores y en general a cualquier persona que esté interesada en mejorar sus habilidades al usar las tecnologías desarrolladas por Microsoft. Esta plataforma ofrece cursos relacionados con tecnologías desarrolladas por Microsoft. Están divididos por módulos y en su mayor parte están compuestos por videos. Los cursos están disponibles en cualquier momento y no tienen plazos para culminarlos. Se realiza una evaluación al finalizar cada módulo. Permite realizarla varias veces hasta que el alumno logre superarla. Permite a los aprendices interactuar por medio de foros. El estudiante puede llevar registro de su progreso mientras realiza el curso. La plataforma está disponible a través de internet y ofrece una interfaz intuitiva, por lo que es muy fácil de usar.
4. COURSEERA: [Coursera, 2016] es una plataforma de educación virtual desarrollada por la Universidad de Stanford, la cual ofrece cursos de diversas áreas. Todo esto respaldado por más de 150 universidades de todo el mundo. Los cursos ofrecidos por esta plataforma pueden ser gratuitos o pagos. Estos se dividen por semanas y tiene plazos establecidos para su realización. Son elaborados e impartidos a través de videos por profesores de una determinada universidad. Se realizan al finalizar la presentación de un concepto determinado. Al final de la semana se debe realizar un trabajo con unos lineamientos establecidos. Permite repetir las evaluaciones y por cada intento se penaliza un porcentaje sobre la nota. Permite a los aprendices interactuar por medio de foros. El estudiante puede llevar registro de su progreso mientras realiza el curso. La plataforma está disponible a través de internet y ofrece una interfaz intuitiva, por lo que es muy fácil de usar. También ofrece una aplicación móvil que permite a los usuarios acceder fácilmente a los cursos.

Las implementaciones antes mencionadas ofrecen una gran accesibilidad a sus funcionalidades y todas cumplen con los requerimientos básicos de un LMS. Sin embargo, estas implementaciones realizan evaluaciones convencionales lo cual dificulta la detección de falencias conceptuales en los estudiantes.

Actualmente existen modelos para diagnosticar el nivel de aprendizaje de un estudiante. [Conejo et al., 2001] propone un modelo para el diagnóstico del estudiante en el que utiliza redes bayesianas. [Jiménez and Ovalle, 2012] describe un modelo de evaluación adaptativa

del nivel de conocimientos del estudiante empleando técnicas propias de la Inteligencia Artificial. [Robles Pedrozo and Rodriguez-Artacho, 2012] propone un modelo de diagnóstico y evaluación basado en la Teoría de Respuesta al Item permitiendo la detección de falencias conceptuales mediante evaluaciones.

Los modelos para al diagnóstico de falencias conceptuales permiten crear plataformas para automatizar dicho proceso. [Brusilovsky and Weber, 2001] presenta ELM-ART, un sistema para el aprendizaje de programación en LISP, el cuál permite una navegación adaptativa de los recursos de aprendizaje de acuerdo a test que se le realizan al estudiante. [FROAC, 2016] es un repositorio de objetos de aprendizaje que se usan como material de estudio y su presentación se adapta de acuerdo a los resultados de las evaluaciones del estudiante.

Por otra parte los LMS pueden ser construidos a partir de ciertas funcionalidades proporcionadas por un conjunto de servicio web relacionados con e-learning. Esto eliminaría ciertas fallas de interoperabilidad entre componenetes que han sido elebarods y ejecutados sobre diferentes plataformas.

Su [Su et al., 2007] propone construir un conjunto de servicios web para e-learning. Los servicios web desarrollados incluyen evaluaciones, administración de cursos, clasificación, calificación, registro y reporte.

Phankokkruad [Phankokkruad and Woraratpanya, 2009] presenta una arquitectura de comunicación basada en servicios web SOAP para mejorar la interoperabilidad entre diferentes LMS. Estos servicios web son usados para definir la estructura de los datos del LMS para que pueda interactuar con sistemas externos.

Fragoso [Fragoso et al., 2014] describe una arquitectura orientada a servicios para LMS y presenta un análisis de características de un conjunto de veinte SGA para propósitos de identificar si son susceptibles de extensión, de tal manera que soporten la integración de Servicios Web de Aprendizaje en lugar de los objetos de aprendizaje.

Partheeban [Partheeban and SankarRam, 2014] propone un LMS construido a partir de servicios web SOAP. Este es un sistema de 3 capas elaborado usando Microsoft ASP.NET 3.5 con VB.NET. Se enfoca en varias características cómo : Administración de contenido, Protección de contenido, Administración de aprendizaje, Administración de evaluaciones, etc.

2.7. Conclusiones

En este capítulo se enuncian los conceptos básicos y características de un Sistema Gestor de Aprendizaje. A su vez, se presenta su evolución a lo largo del tiempo y soluciones propuestas para adaptar el entorno y las rutas de aprendizaje de los estudiantes. Además, se realiza un comparativo de implementaciones de LMS en donde se describen sus estructuras, sus cursos y sus estrategias de evaluación. Por último, se enuncian trabajos y soluciones de LMS encontrados en la literatura, los cuáles permiten evidenciar el problema planteado y justificar la solución propuesta.

3 Metodología

3.1. Introducción

Una vez planteados los objetivos que delimitan el alcance de este trabajo de investigación, la definición de los conceptos claves y análisis de trabajos en esta línea que se han realizado en los últimos años se presenta a continuación la metodología descrita a través de 4 fases que permitieron el logro de los objetivos propuestos.

3.2. Plan de Trabajo

Fase 1: Análisis y comparación de repositorios de Sistemas Gestores de Aprendizaje.

Esta fase comprende el cumplimiento de los objetivos 1 y 2, a través de las siguientes actividades:

- Definir la estructura de los LMS y de sus cursos.
- Identificar las implementaciones de LMS existentes y definir tanto sus atributos como los de sus cursos.
- Cotejar las características de los LMS, tales como visualización del contenido, flexibilidad en la creación y acceso de los cursos.
- Identificar y analizar las características de los cursos, tales como personalización y presentación del material temático y método de evaluación.
- Determinar las bases de diseño de la plataforma.

Fase 2: Selección de herramientas y diseño de estructura para la implementación del sistema.

Esta fase comprende el cumplimiento de los objetivos 3 y 4, a través de las siguientes actividades:

- Comparar y seleccionar las herramientas que permitan la creación y el soporte de la plataforma que se propone, teniendo en cuenta características como personalización y presentación de la información, y flexibilidad en el desarrollo de la plataforma.

- Definir los módulos del sistema y su estructura de interacción, en función del diseño establecido en la fase 1.

Fase 3: Implementación y validación del prototipo propuesto.

Esta fase comprende el cumplimiento de los objetivos 5 y 6, a través de las siguientes actividades:

- Elaborar los módulos del sistema acorde a la estructura definida en la fase anterior.
- Establecer el mecanismo de conexión entre los módulos.
- Integrar el mecanismo de evaluación y diagnóstico como recurso del LMS.
- Creación y montaje de un curso modelo para validar el sistema.

Fase 4: Documentación

Elaboración del documento de trabajo de grado.

4 Implementación

4.1. Introducción

En este capítulo inicialmente se describe el diseño e implementación del motor de diagnóstico. Además, se presenta la arquitectura utilizada para el desarrollo de la solución propuesta y posteriormente se describe en detalle la implementación de la solución. Finalmente se presenta el servicio web (REST) construido.

4.2. Diseño e Implementación del Motor de Diagnóstico

[Robles Pedrozo and Rodriguez-Artacho, 2012] propone un modelo de evaluación y diagnóstico mediante test aplicados a estudiantes, y que combina la Teoría de Respuesta al Ítem para generar -dentro de una escala-, el nivel en el que se encuentra el conocimiento del evaluado. A partir de esta información, clasificarlos de acuerdo con el nivel en la escala, y adicionalmente generar un diagnóstico personalizado que ayude a los estudiantes a identificar cuáles son los conceptos que desconocen y las posibles rutas de aprendizaje para mejorar su rendimiento.

4.2.1. Arquitectura del modelo

En la Figura 4-1 se ha representado la arquitectura del modelo de evaluación y diagnóstico que se propone en este proyecto. En esta arquitectura se pueden distinguir los siguientes componentes:

- a **Módulo experto:** Contiene el conocimiento aportado por el experto, en este caso el profesor. Este módulo se compone a su vez de dos partes: La primera, un modelo o mapa conceptual que representa el dominio de la asignatura, e incluye conceptos y las relaciones jerárquica (o de dependencia) entre éstos; la segunda, un banco de ítems a través de los cuales se construyen los test, se evalúan los conceptos y son los presentados al estudiantes que va a ser evaluado.
- b **Módulo de Construcción y Especificaciones del test:** Los tests se construyen a partir de los ítems registrados en el banco de ítems, sin embargo, son test pre-definidos, es decir, los ítems que van a formar parte de un test cuando son ingresados, llevan un campo bandera que indica a qué test van a pertenecer. Lo anterior, se hace con el fin de

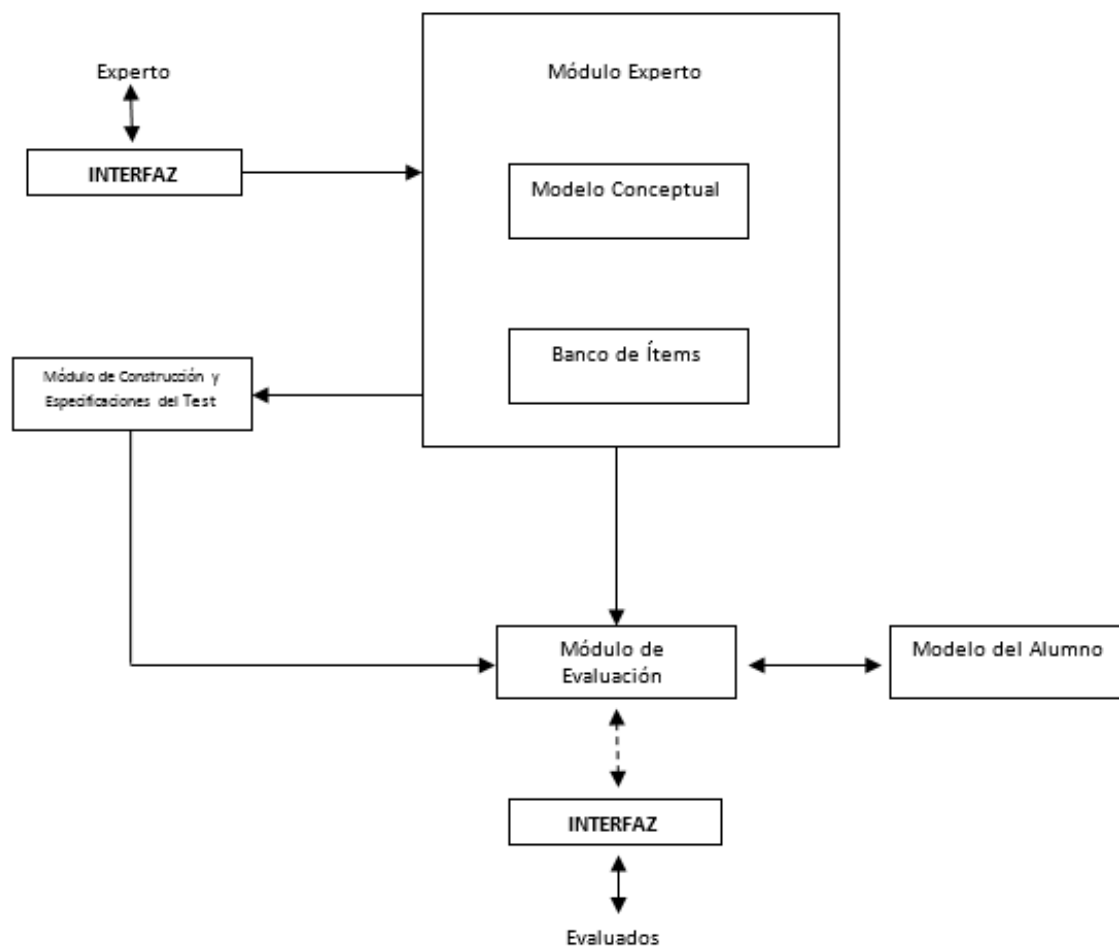


Figura 4-1: Arquitectura del Modelo de Evaluación y Diagnóstico

garantizar que los test ya han sido aplicados en campo, son fiables y válidos. Este módulo permite al docente:

- Insertar las preguntas que corresponden a un test.
- Especificar qué conceptos son evaluados en cada pregunta.
- Especificar la relación entre conceptos, o lo que es lo mismo, estructuración de la temática a evaluar.

c **Modelo del alumno:** Se encarga de almacenar las respuestas que el estudiante ha dado a las preguntas del test, y es fundamental para el diagnóstico (estimación del nivel de habilidad e identificación del camino de aprendizaje). Inicialmente, del estudiante se obtienen solo sus datos básicos (identificación, nombre, grado y grupo), pero posteriormente, se almacena su vector de respuestas, su nivel de conocimiento, el grupo en el que queda clasificado, y una descripción del camino de conceptos en los que debe reforzar porque presenta falencias y por tanto debe mejorar. Conceptos es el término con el que

se hace referencia a la estructura jerárquica de la asignatura. Ver Figura 4-2.

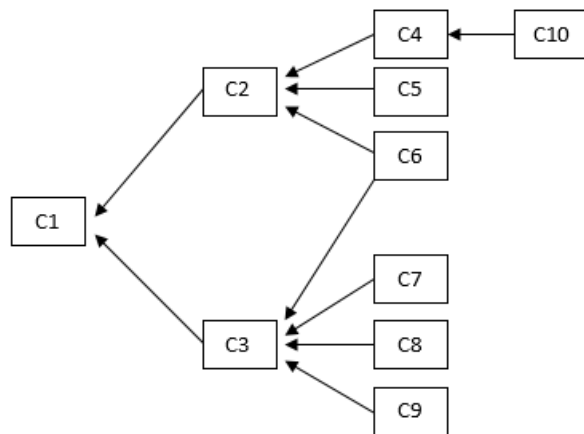


Figura 4-2: Estructura Jerárquica – Relación entre Conceptos

d **Módulo de evaluación:** Este módulo contiene la escala de evaluación del conocimiento, y se encarga de:

- Hacer el cálculo de probabilidad de acierto de cada pregunta.
- Hacer el cálculo del nivel de habilidad de cada estudiante.
- Generar el diagnóstico del evaluado.

Es a través de este módulo que se utiliza de evaluación basado en un Modelo de Relación Temas – Conceptos o entre conceptos, y en un enfoque estadístico, el cual permite representar las relaciones prerrequisitos que se pueden presentar entre los conceptos pertenecientes a una estructura jerárquica conceptual.

e **Módulo de Presentación (Interfaz):** En este módulo interactúa el experto (docente, tutor, profesor) con el sistema, así como el evaluado con el sistema. El experto interviene para: describir la estructura del modelo conceptual de la temática a evaluar, ingresar el conjunto de ítems que evalúan la temática y a su vez conforman un test, y para la construcción y especificaciones particulares de cada test. El evaluado interactúa con el sistema porque a través de él le es presentado el conjunto de ítems -amarrados a conceptos-, que se pretenden evaluar, y a partir de ahí generar el diagnóstico.

4.2.2. Modelo de Relación entre Temas y Conceptos

Un Modelo de Relación Concepto-Efecto fue propuesto inicialmente por [Hsu and Gupta, 1998], en el que demostraba cómo el nivel de conocimiento de ciertos conceptos podían ser influenciados por el nivel de conocimientos de otros conceptos. El modelo fue aplicado sobre los

resultados de evaluaciones de estudiantes, y éstos recibieron orientación en cuando a qué áreas se necesitaban mejorar y cómo podría ser mejorado el nivel de conocimiento.

Más adelante Hwang ha tomado como base el Modelo de Relación Concepto-Efecto demostrando una vez más que el aprendizaje eficaz de conceptos requiere generalmente el conocimiento de conceptos básicos [Hwang, 2003], [Hwang and Hsiao, 2003]; e incluso han desarrollado herramientas para asistir a docentes en la definición de tales relaciones [Hwang, 2005]. Para detectar qué conceptos no se tienen claros, se hace uso de un Modelo de Propagación Concepto-Efecto propuesto por [Chu et al., 2006], este enfoque es usado para modelar la estructuración jerárquica de una temática y de esta forma representar la relación entre temas, subtemas y conceptos, que para el caso de esta investigación todas las definiciones de títulos (temas y subtemas) serán llamadas ‘Conceptos’.

El Modelo de Propagación Concepto-Efecto registra todos los conceptos que pueden ser influenciados por cada concepto en el proceso de aprendizaje. En la Figura 4-3 se presenta un ejemplo de la Relación Concepto-Efecto para el componente de Aleatoriedad en Matemáticas.

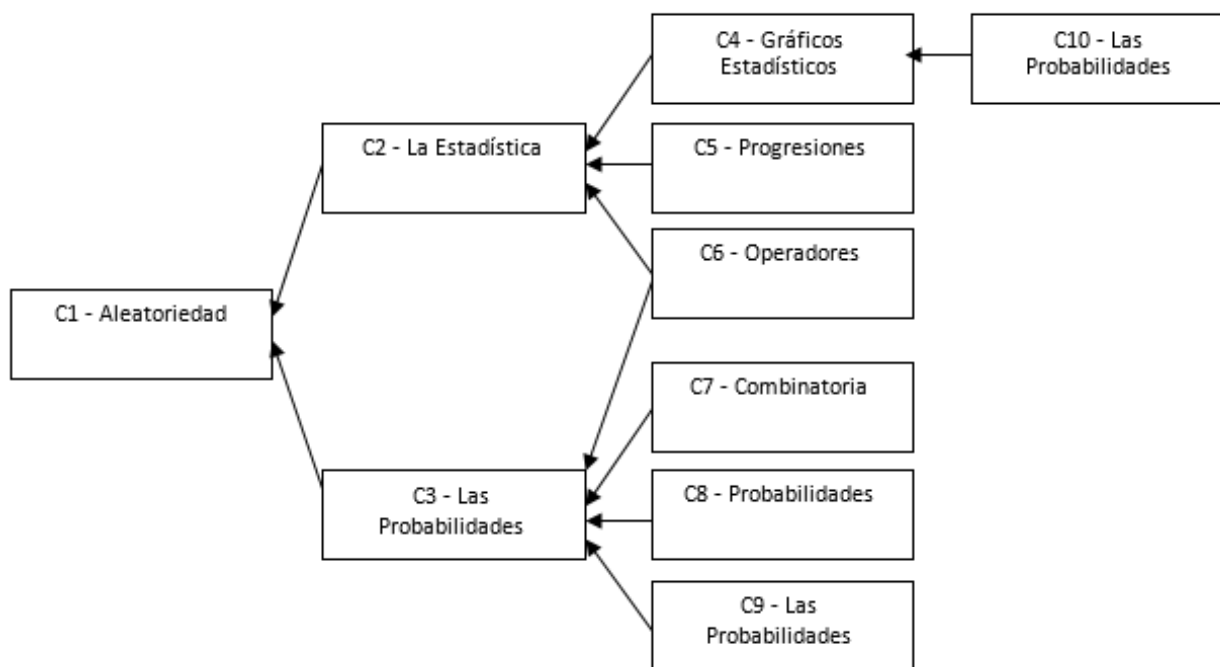


Figura 4-3: Relación entre Conceptos – Diagrama Jerárquico Aleatoriedad en Matemáticas

Los conceptos que afectan al concepto C1 son: C2 y C3 (La Estadística y Las Probabilidades); los conceptos que afectan al concepto C3 son: C6, C7, C8 y C9; los conceptos que afectan a C2 son: C4, C5, C6 y C10; pero que afecten solo directamente al concepto C2, son: C4, C5 y C6.

A partir del diagrama que se presenta en la Figura 4-3, el Modelo de Propagación Concepto-Efecto se concreta en la Figura 4-4, y se lee como: los conceptos C_j afectados por los

conceptos C_i ; o lo que es lo mismo, a qué conceptos C_j , los conceptos C_i afectan. Se representa el cruce (C_i, C_j) con un '1' a los conceptos que tienen relación, y con un '0' los conceptos que no están relacionados. De acuerdo con la Figura 4-3, si un estudiante falla en el aprendizaje de un concepto C_i y lo evidencia en proceso de evaluación, no está preparado ni conoce los conceptos C_j que son afectados por C_i .

		Conceptos - Cj									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Conceptos - Ci	C1	1	0	0	0	0	0	0	0	0	0
	C2	1	1	0	0	0	0	0	0	0	0
	C3	1	0	1	0	0	0	0	0	0	0
	C4	1	1	0	1	0	0	0	0	0	0
	C5	1	1	0	0	1	0	0	0	0	0
	C6	1	1	1	0	0	1	0	0	0	0
	C7	1	0	1	0	0	0	1	0	0	0
	C8	1	0	1	0	0	0	0	1	0	0
	C9	1	0	1	0	0	0	0	0	1	0
	C10	1	1	0	1	0	0	0	0	0	1

Figura 4-4: Tabla Relación entre Conceptos - Conceptos C_j afectados por C_i

Se deben crear o implementar dos estructuras (C_i, C_j) en las que se registre:

- Primera estructura: Los conceptos C_j (primera fila de títulos superior) afectados por C_i (primera columna de títulos a la izquierda).
- Segunda estructura. Los conceptos C_j (primera fila de títulos superior) afectados directamente por C_i (primera columna de títulos a la izquierda).

La primera estructura se crea con el fin de conocer cuáles son los conceptos C_j -de nivel superior-, que el concepto C_i afecta, y que por tanto, en el evento en que este concepto C_i no haya sido superado por un estudiante, se puedan tener identificados los conceptos C_j que son afectados por la debilidad que se tiene en el dominio del concepto C_i .

La segunda estructura permite conocer cuáles son los conceptos C_i -de nivel inferior-, que afectan directamente al concepto C_j , y que por tanto, en el evento en que este concepto C_j no haya sido superado por un estudiante, pueda indagarse por los conceptos previos C_i que el estudiante debe dominar para que esté en condiciones de superar el concepto C_j . Ver Figura 4-5 .

Las dos estructuras van a permitir generar, por estudiante, la/las rutas de aprendizaje a través de conceptos en los que se haya evidenciado las fallas o desconocimiento.

		Conceptos - Cj									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Conceptos - Ci	C1	1	0	0	0	0	0	0	0	0	0
	C2	1	1	0	0	0	0	0	0	0	0
	C3	1	0	1	0	0	0	0	0	0	0
	C4	0	1	0	1	0	0	0	0	0	0
	C5	0	1	0	0	1	0	0	0	0	0
	C6	0	1	1	0	0	1	0	0	0	0
	C7	0	0	1	0	0	0	1	0	0	0
	C8	0	0	1	0	0	0	0	1	0	0
	C9	0	0	1	0	0	0	0	0	1	0
	C10	0	0	0	1	0	0	0	0	0	1

Figura 4-5: Tabla Relación entre Conceptos - Conceptos C_i que afectan DIRECTAMENTE a C_j

4.2.3. Funcionalidad del modelo

Identificar desde el punto de vista gráfico cuáles son temas, subtemas y conceptos, en una estructuración jerárquica, resulta sencillo; sin embargo, llevar ésto a un sistema para que de manera automática se pueda conocer qué conceptos se evalúan en un test, qué conceptos se evalúan en cada pregunta, en qué proporción cada concepto es evaluado en cada pregunta, resulta un poco complejo; de ahí que los investigadores en esta línea, hayan hecho uso, por ejemplo, de un enfoque estadístico para aplicar a un conjunto de evaluaciones [Hwang and Hsiao, 2003], y probar que tan efectivo resulta ser este registro de relaciones entre conceptos, y que tan eficiente resulta ser para el docente, en cuanto a poder identificar caminos de aprendizaje.

Adicional al planteamiento realizado en la sección C2, y para ver cómo opera el modelo, se va a suponer que se cuenta con un test de 10 ítems o preguntas, identificadas así: $I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}$. Se genera una tabla que representa la relación que hay entre ítems y conceptos. En la tabla, a cada pregunta se le relaciona el concepto o los conceptos que son evaluados, y en qué proporción [Hwang, 2007], es decir, si un concepto pesa más que otro dentro de la pregunta. Cada valor almacenado en la tabla está representado con un número entero entre cero '0' y cinco '5' [Hwang, 2007]:

0. No hay relación
1. Muy débil relación
2. Débil relación

3. Media relación
4. Fuerte relación
5. Muy fuerte relación

Son los expertos, docentes, responsables del desarrollo de las temáticas, los encargados de asignar a cada pareja (I_i, C_j) el valor que representa la influencia de uno o varios conceptos dentro de una pregunta o ítem.

Para representar el peso total de un concepto evaluado dentro de un examen, se calcula la variable $TPCI(C_j)$, como la $\sum_{i=1}^n (I_i, C_j)$ ('n' representa el número de preguntas, y 'j' el concepto evaluado).

Para el caso de respuesta incorrectas a preguntas que de igual forma están evaluando conceptos, se calcula una variable $R_i(C_j)$, que representa el peso total de estos conceptos evaluados incorrectamente.

$$R_i(C_{j=1}) = \sum(I_i, C_j)$$

El subíndice 'i' toma los valores del número de la pregunta que fue respondida de manera errónea. El subíndice 'j' toma el valor de 1, como ejemplo para indicar que el concepto 1 es el que se está evaluando dentro del test, y es el concepto al que se le está calculando el peso total de respuestas incorrectas.

Finalmente, la proporción de respuestas correctas relevantes a un cierto concepto C_j $PRc(C_j)$, se calcula como:

$$PRc(C_{j=1}) = \frac{[TPCI(C_j) - R_i(C_j)]}{TPCI(C_j)}$$

La igualdad (j=1) representa que el concepto evaluado, en este caso, es el concepto 1. $PRc(C_j)$ es la variable que indica el grado de importancia de un concepto dentro de un examen específico, preparado para evaluar un tema.

La Figura 4-6, presenta a modo de ejemplo, una representación de la relación Ítems-Conceptos de un tipo de examen.

En la Figura 4-6, el cálculo de $R_i(C_j)$ se hizo bajo el supuesto de que las respuestas a los ítems I3, I6, e I7 fueron incorrectas. La Figura 4-7 muestra el el vector de respuestas del estudiante para este caso.

El valor cero '0' en una casilla representa des-acierto en la respuesta; un valor de uno '1', representa acierto en la respuesta.

Dejar la tarea de asignar el peso o la importancia de un concepto dentro de un ítem, en manos de un solo experto ha causado controversia, y ha sido razón para que, inclusive en esa línea, se investiguen metodologías que permita la imparcialidad y este valor sea ajeno a la opinión de un solo experto [Panjaburee et al., 2009]; sin embargo, en esta investigación se va a partir de la opinión del experto en la temática que se quiere evaluar de manera particular.

		Conceptos - C _j									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Ítem - I _i	I1	5	1	0	0	0	0	0	0	0	0
	I2	0	4	2	0	0	0	0	0	0	0
	I3	0	0	3	1	2	0	0	0	0	0
	I4	0	0	0	5	0	0	0	0	0	0
	I5	0	0	0	0	5	0	0	0	0	0
	I6	1	0	0	0	0	4	1	0	0	0
	I7	0	0	0	0	0	0	5	0	0	0
	I8	0	0	0	0	0	0	0	3	1	2
	I9	0	0	0	0	0	1	0	0	4	0
	I10	0	0	0	0	0	1	0	2	0	5
TP C _i	6	5	5	6	7	6	6	5	5	7	
R _i	1	0	3	1	2	4	6	0	0	0	
PR _c	0,83	1	0,4	0,83	0,71	0,33	0	1	1	1	

TP C_i Total del Peso del Concepto 'i' sobre los Ítems de un test
R_i Respuestas Incorrectas
PR_c Proporción de Respuestas correctas por cada concepto

Figura 4-6: Relación Items-Conceptos

A partir de los resultados presentados en la Figura 4-6, más exactamente el resultado de la variable $PR_c(C_j)$, la Figura 4-8 muestra la estructura jerárquica de una temática, con los pesos totales de cada concepto dentro de una evaluación

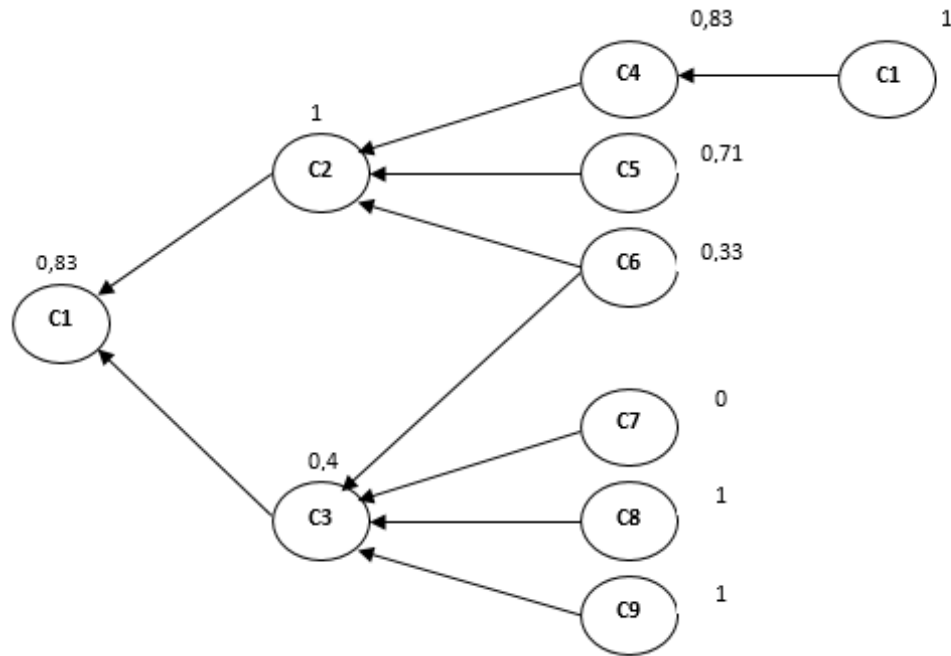
De la información presentada en la Figura 4-8, se puede inferir que:

- Los conceptos C2 y C3 son fundamentales para el aprendizaje del concepto C1.
- Los conceptos C6, C7, C8, C9, son fundamentales para el aprendizaje del concepto C3.
- Los conceptos C3, C4, C5, C10, son fundamentales para el aprendizaje del concepto C2.
- El concepto C10 es fundamental para el aprendizaje del concepto C4.

Para el diagnóstico de los problemas de conocimiento de ciertos conceptos, es fundamental tener identificados todas las posibles rutas de aprendizaje. Esta información se puede obtener directamente del gráfico de la Figura 4-8; sin embargo, se pueden obtener de la lectura que se haga de la Figura 4-4.

<u>Nom.</u>	<u>I1</u>	<u>I2</u>	<u>I3</u>	<u>I4</u>	<u>I5</u>	<u>I6</u>	<u>I7</u>	<u>I8</u>	<u>I9</u>	<u>I10</u>
<u>STU</u>	1	1	0	1	1	0	0	1	1	1

Figura 4-7: Vector Respuestas Estudiante

Figura 4-8: Estructura Jerárquica de Conceptos con el valor asignado $PRc(C_j)$

A partir de la información contenida en la Tabla III-5, las rutas o caminos de aprendizaje son los siguientes:

- R1: C1
- R2: C2 → C1
- R3: C3 → C1
- R4: C4 → C2 → C1
- R5: C5 → C2 → C1
- R6: C6 → C3 → C2 → C1
- R7: C7 → C3 → C1
- R8: C8 → C3 → C1
- R9: C9 → C3 → C1
- R10: C10 → C4 → C2 → C1

En el caso de la Ruta 6, se lee: Para conocer el concepto C1, desde el concepto C6, se deben afianzar los conocimientos de los conceptos $C6 \rightarrow C3 \rightarrow C2 \rightarrow C1$. Si bien los conceptos C2 y C3 no están conectados, el modelo que se plantea en esta investigación considera no solo la

ruta de aprendizaje, sino también los conceptos previos que se requieren conocer o aprender para poder determinar si un concepto de nivel superior es conocido o no.

Teniendo en cuenta que se está trabajando con un enfoque estadístico, y a partir de los resultados presentados en la Figura III-4, es evidente que algunos valores de la variable $\text{PRc}(C_j)$ están muy bajos con respecto a los otros. Se hace necesario definir un valor umbral (u) que represente el mínimo valor de proporción de respuesta correcta permitido, un valor $\text{PRc}(C_j)$ superior a ese umbral indica que el estudiante conoce y ha aprendido el concepto evaluado, un valor $\text{PRc}(C_j)$ inferior, sugiere que el estudiante desconoce o no ha aprendido el concepto.

Para el caso de esta investigación se toma el valor del umbral (u) como 0.7. Siendo así, se tiene que los conceptos $\text{PRc}(C3)$, $\text{PRc}(C6)$, y $\text{PRc}(C7)$ están por debajo de 0,7.

$$\text{PRc}(C3) = 0.4$$

$$\text{PRc}(C6) = 0.33$$

$$\text{PRc}(C7) = 0$$

Para determinar las Rutas que el estudiante debe reforzar, se tienen en cuenta:

- Primer caso: Los conceptos que se afectan a partir del concepto identificado con falencia o que no alcanzó el umbral de 0.7.
- Segundo caso: Los conceptos previos que el estudiante tendría que reforzar para que el concepto con falencia sea superado.

Lo siguiente es identificar las posibles rutas de aprendizaje sabiendo cuáles son los conceptos que no alcanzaron el umbral de 0.7.

Primer caso

1. A qué conceptos afecta el concepto C3?

Se hace un recorrido de la Tabla que aparece en la Figura 4-4, por fila, y desde el concepto fallido, en este caso C3. El resultado es que afecta al concepto C1.

Si $\text{PRc}(C1) > 0,7$ entonces

Ruta1 \leftarrow (C3)

Sino

Si $\text{PRc}(C1) < 0,7$ entonces

Ruta1 \leftarrow (C3 \rightarrow C1)

FSi

FSi

Resultado: Ruta1 \leftarrow (C3)

2. A qué conceptos afecta el concepto C6?

Se hace un recorrido por la Tabla que aparece en la Figura 4-4, por fila, justo desde el concepto con falencia, en este caso C6. C6 afecta a los conceptos C3, C2, y C1.

```

Ruta[1,1] = C6
Si PRc(C3) < 0.7 entonces
Ruta[1,2] = C3
Si PRc(C2) < 0.7 entonces
Ruta[1.3] = C2
Si PRc(C1) < 0,7 entonces
Ruta[1.4] = C1
Ruta2 <-- (C6 --> C3 --> C2 --> C1)
Sino
Ruta2 <-- (C6 --> C3 --> C2)
FSi
Sino
Ruta2 <-- (C6 --> C3)
FSi
Sino
Ruta2 <-- (C6)
FSi
Resultado: Ruta2 <-- (C6 --> C3)

```

Independientemente de si en el grafo, los conceptos C2 y C3 no están en la misma ruta de aprendizaje, como ya se comentó en párrafos anteriores, ambos conceptos son afectados por el concepto C6 y si se cumple que $C6 < 0,7$ y $C3 < 0,7$ y $C2 < 0,7$ entonces es válida y posible la ruta de aprendizaje $C6 \rightarrow C3 \rightarrow C2$.

3. A qué conceptos afecta el concepto C7?

Se hace un recorrido por la Figura 4-4, por fila, justo desde el concepto con falencia, en este caso C7. C6 afecta a los conceptos C7, C3, y C1.

```

Ruta[1,1] = C7
Si PRc(C3) < 0.7 entonces
Ruta[1.2] = C3
Si PRc(C1) < 0.7 entonces
Ruta[1.3] = C1
Ruta3 <-- (C7 --> C3 --> C1)
Sino
Ruta3 <-- (C7 --> C3)
FSi

```

```

Sino
Ruta3 <-- (C7)
FSi
Resultado: Ruta3 <-- (C7 --> C3)

```

Las rutas o caminos de aprendizaje que aplican para este caso son:

```

Ruta1 ← (C3)
Ruta2 ← (C6 → C3)
Ruta3 ← (C7 → C3)

```

Segundo caso

1. Qué conceptos afectan directamente al concepto C3?

Se hace un recorrido por la Tabla que aparece en la Figura 4-5, por columna, organizando parejas de conceptos ($C_i \rightarrow C_j$) desde ($i=10$ -último de los conceptos-), hasta el concepto con falencia, así:

Si $(C_i, C_j) = 1$, con $(C_j = C_3)$, entonces se debe reforzar el camino $C_i \rightarrow C_3$; y así sucesivamente hasta que $i=3$, o lo que es lo mismo $C_i = C_3$.

Para el caso del ejemplo que se está trabajando, los conceptos que afectan al concepto C3 son: C6, C7, C8, y C9. Al hacer el recorrido por columna, justo la columna del concepto C3, las parejas que se organizarían como rutas de aprendizaje son:

```

R1: C6 → C3
R2: C7 → C3
R3: C8 → C3
R4: C9 → C3

```

En este punto solo se tiene los conceptos que afectan directamente al concepto C3, pero no, si estos conceptos tienen falencias, de ser así, las posibles rutas de aprendizaje quedarían confirmadas.

```

Si  $PR_c(C6) < 0,7$  entonces
Ruta1 <-- (C6 --> C3)
FSi

```

De igual forma para C7, C8, C9.

2. Qué conceptos afectan directamente al concepto C6?

Se hace un recorrido por la Tabla que aparece en la Figura 4-5, por columna, organizando parejas de conceptos ($C_i \rightarrow C_j$) desde ($i=10$ -último de los conceptos-), hasta el concepto con falencia, así:

Si $(C_i, C_j) = 1$, con $(C_j = C6)$, entonces se debe reforzar el camino $C_i \rightarrow C6$; y así sucesivamente hasta que $i=6$, o lo que es lo mismo $C_i = C6$. En este caso, todos los

valores almacenados en las celdas (C_i, C_j) son iguales a cero, por lo que entonces se concluye que el concepto C6 no tiene pre-requisitos.

Ruta2 \leftarrow (C6)

3. Qué conceptos afectan directamente al concepto C7?

Se hace un recorrido por la Tabla que aparece en la Figura 4-5, por columna, organizando parejas de conceptos $(C_i \rightarrow C_j)$ desde $(i=10$ -último de los conceptos-), hasta el concepto con falencia, así: Si $(C_i, C_j) = 1$, con $(C_j = C7)$, entonces se debe reforzar el camino $C_i \rightarrow C7$; y así sucesivamente hasta que $i=7$, o lo que es lo mismo $C_i = C7$. En este caso, todos los valores almacenados en las celdas (C_i, C_j) son iguales a cero, por lo que entonces se concluye que el concepto C7 no tiene pre-requisitos.

Ruta3 (C7)

El Segundo caso se implementa como una manera de corroborar las rutas identificadas en el Primer caso, y solo si son conceptos terminales, se confirman como nodos básicos de aprendizaje.

4.2.4. Implementación del Sistema de Evaluación y Diagnóstico del Aprendizaje

[Robles Pedrozo and Rodriguez-Artacho, 2012] propone un sistema de evaluación y diagnóstico con enfoque estadístico y un modelo de relación entre conceptos, se requiere de varias interfaces y algoritmos que permiten realizar los cálculos y recorridos para realizar una retro-alimentación inteligente. En la Figura 4-9 se muestra el diagrama de funcionamiento del módulo de diagnóstico; y en la Figura 4-11 se presenta el funcionamiento general del modelo propuesto.

El sistema de evaluación y diagnóstico del aprendizaje propuesto por [Robles Pedrozo and Rodriguez-Artacho] fue implementado para el actual trabajo como una librería para el lenguaje de programación JAVA. La librería recibe como nombre DIAGEN (Por sus siglas en inglés Diagnostic Engine). DIAGEN fue elaborada bajo el patrón de diseño STATE. La Figura 4-10 muestra el flujo de estados de DIAGEN. A continuación se explican cada uno de los estados:

- Agregar conceptos: Se agregan los concepto que son evaluados en el Test.
- Agregar enlaces entre conceptos: Se define la forma en que están relacionados los conceptos. Esto es determinado por el experto.
- Agregar respuestas: Se ingresan el número de respuestas del test y las respuestas del estudiante a ser evaluado.

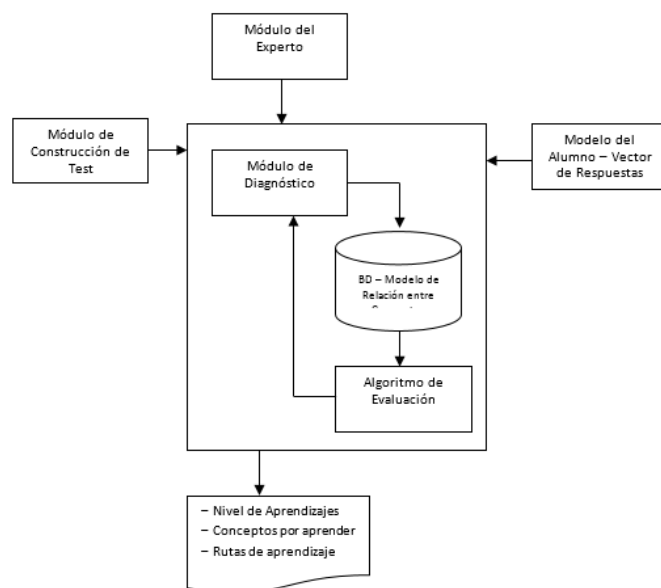


Figura 4-9: Funcionamiento del Módulo de Diagnóstico

- Agregar pesos de conceptos por items: Se determina que peso tiene cada concepto sobre cada uno de los items del test. Los valores a ingresar los determina la escala que se nombró en la sección anterior.
- Agregar patrones: Este estado es opcional. Se puede realizar el diagnóstico sin llegar a este estado. Sirve para comparar patrones de respuestas de los estudiantes.

Por otra parte la Figura 4-12 muestra el diagrama de clases del motor de diagnóstico. La interface Stat define todos los metodos que pueden implementar los estados, mientras que las interface Automaton define los métodos que manejan el flujo de estados. Los estados explicados anteriormente están implementados en las siguientes clases:

- AddConceptState: Implementa el estado agregar conceptos.
- AddEdgeState: Implementa el estado agregar enlaces entre conceptos.
- AddAnswerState: Implementa el estado agregar respuestas.
- AddWeightState: Implementa el estado agregar pesos de conceptos por item.
- AddPatternsState: Implementa el estado agregar patrones.

Anéxo a este documento se encuentra el código fuente de la librería implmentada.

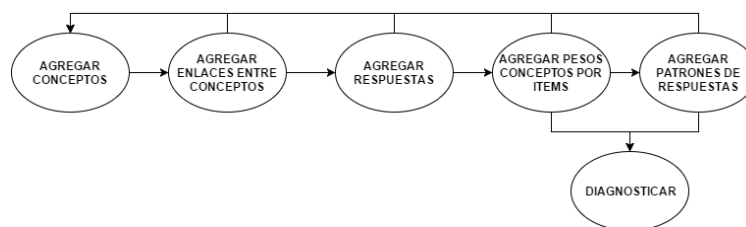


Figura 4-10: Flujo de Estados DIAGEN

4.3. Diseño e Implementación de la Arquitectura

La construcción del prototipo de LMS se hizo bajo una arquitectura de capas. Cada proceso compone una capa que puede describirse en función de un conjunto de responsabilidades bien marcadas.

La Figura 4-13 muestra la arquitectura propuesta. En ella se aprecian 4 capas las cuales se describen a continuación:

4.3.1. Capa de datos

Su responsabilidad consiste en proveer un mecanismo de persistencia que permita almacenar, consultar, modificar, y eliminar aquellos datos queya no nos interese. Esta capa se implementa con una base de datos, en nuestro caso seleccionamos MySQL.

4.3.2. Capa de objetos de acceso a los datos (DAO)

Esta capa interactua directamente con la capa de datos y es responsable de proveer objetos que encapsulen y resuelvan cualquier tipo de acceso de acceso u operación relacionada con los datos a los que la aplicación debe acceder. Dentro de los métodos de estos objetos , se escribe y ejecuta las sentencias SQL. Para su implementación se utiliza el framework Hibernate, el cuál sirve para describir el esquema de la capa de datos y el mapeo de los objetos en la base de datos. SU implmentación se da en los paquetes *org.utb.shirepo.model* y *org.utb.shirepo.dao*.

4.3.3. Capa de negocios

Esta capa tiene la responsabilidad de procesar y exponer las reglas del negocio de la aplicación. Se utiliza el framework Spring para realizarla, este framework proporciona inyección de dependencias y permite integrar los frameworks Hibernate y Jersey, encargados de el mapeo de los objetos y de la implementación de los servicios web respectivamente.

4.3.4. REST API

En esta capa se encuentra la implementación de la API REST. Los servicios web prporcionan las funcionalidades básicas del LMS. Esta API fue elaborada con el framework JERSEY, el cuál proporciona una implementación de JAX-RS, una API del lenguaje de programación JAVA que proporciona soporte para la creación de servicios web REST. La Sección 2.4.1 explica en detalle esta capa.

4.4. Diseño e Implementación del Servicio Web

4.4.1. Recursos/URIs

Prefijo URI: shirepo/webapi/*

CONCEPTOS:

GET

/courses/{courseId}/structure/units/{unitId}/concepts

/instructors/{email}/courses/{courseId}/structure/units/{unitId}/concepts

POST

instructors/{email}/courses/{courseId}/structure/units/{unitId}/concepts

Ejemplo:

URI: instructors/juan.ramirezpadilla/courses/2/structure/units/43/concepts

Body:

```
content-type: application/json
{
  "nodeName" : "Clases y Objetos",
  "parent1" : "Fundamentos",
  "parent2": "Paradigma"
}
```

Nota: "nodeName" corresponde al nombre del concepto. Por su parte, "parent1" corresponde al nombre de primer padre (concepto o unidad), y "parent2" corresponde al nombre del segundo padre (concepto o unidad). Los valores ingresados en "parent1" y "parent2" deben pertenecer a conceptos/unidades que se hayan previamente registrado.

A tener en cuenta:

- Un concepto NO puede tener un valor de "parent1" igual al valor de "parent2"
- Un concepto sólo podrá tener como padres a: Una unidad, un concepto, o dos conceptos.

CURSOS:**GET**

- /courses DS
- /courses/{courseId} DS
- /instructors/{email}/courses DS
- /instructors/{email}/courses/{courseId} DS

POST

- /instructors/{email}/courses DS

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses

Body:

content-type: application/json

```
{
  "courseName": "POO",
  "description": "El objetivo de este curso es explicar los conceptos básicos de
  la POO.",
  "status": false
}
```

Nota: Se requiere que el {email} proporcionado en la URI, pertenezca a un Instructor existente en la base de datos. Al crearse un nuevo curso, se crean también una nueva sesión y estructura, exclusivas de dicho curso. Los IDs de la sesión y estructura, serán consecutivos al ID del curso, esto es, si el curso tiene asignado el ID "2", la sesión tendrá asignado el ID "3" y la estructura el ID "4". El campo "status" corresponde al estado de edición ("true" si el curso aún no termina de crearse, o está siendo editado, "false" si la creación/edición del curso ha finalizado).

- /courses/{courseId}/register DS

Ejemplo:

URI: shirepo/webapi/courses/2/register

Body:

content-type: application/json

```
{
  "id" : 5
}
```

Nota: El ID que se envía en el body, corresponde al ID del estudiante que será

registrado en el curso. El estudiante debe existir en la base de datos.

- /courses/{courseId}/unregister

PUT

- /instructors/{email}/courses/{courseId} DS

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2

Body:

content-type: application/json

```
{
  "courseName": "Math",
  "status": false
}
```

DELETE

- /instructors/{email}/courses/{courseId} DS

ESTUDIANTES:

GET

- /students

- /students/{email}

- /instructors/{email}/courses/{courseId}/students/registered

POST

- /students

Ejemplo:

URI: shirepo/webapi/students

Body:

content-type: application/json

```
{
  "userEmail": "jorgeramirezp96@gmail.com",
  "userLastName": "Ramirez",
  "userName": "Jorge",
  "password": "password"
}
```

Nota: Solo se requieren estos campos para crear un nuevo instructor.

PUT

- /students/{email}

Ejemplo:

URI: shirepo/webapi/students/jorgeramirezp96

Body:

content-type: application/json

```
{
  "userEmail": "jorgeramirezp96@gmail.com",
  "userLastName": "Ramirez",
  "userName": "Jorge Mario",
  "password": "password"
}
```

DELETE

- /students/{email}

INSTRUCTORES:**GET**

- /instructors DS

- /instructors/{email} DS

POST

- /instructors DS

Ejemplo:

URI: shirepo/webapi/instructors

Body:

content-type: application/json:

```
{
  "userEmail": "juan.ramirezpadilla@gmail.com",
  "userLastName": "Ramirez",
  "userName": "Juan",
  "password": "password"
}
```

Nota: Solo se requieren estos campos para crear un nuevo instructor.

PUT

- /instructors/{email} DS

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla

Body:

contenttype: application/json:

```
{
  "userEmail": "juan.ramirezpadilla@gmail.com",
  "userLastName": "Ramirez",
  "userName": "Juan Camilo",
  "password": "password"
}
```

DELETE

- /instructors/{email} DS

ITEMS:

GET

- /instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/items

POST

/instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/items

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/units/7/test/items

Body:

content-type: application/json

```
{
  "name" : "POO_test_item1",
  "statement" : "Que propiedades presenta el nivel de acceso Private?"
}
```

Nota: El campo "name", corresponde al nombre identificador del item. El campo "statement", corresponde al enunciado de la pregunta.

PUT

- /instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/items/{itemId}

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/units/7/test/items/9

Body:

content-type: application/json

```
{
  "name" : "POO_test_item1",
  "statement" : "NO es una propiedad del nivel de acceso Private"
}
```

Nota: El campo "name", corresponde al nombre identificador del item. El campo "statement", corresponde al enunciado de la pregunta.

DELETE

- /instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/items/{itemId}

PESOS:

GET

-/instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/items/{itemId}/weights

POST

-/instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/items/{itemId}/weights

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/units/5/test/items/7/weights

Body:

content-type: application/json

```
{
  "weight" : 5,
  "cid": 3
}
```

Nota: El campo "weight", corresponde al peso asignado al item con respecto a un concepto, este valor debe ser un entero entre 0 y 5. El campo "cid" corresponde al id asignar el peso.

PUT


```
-/instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/
items/{itemId}/weights
```

Ejemplo:

```
URI:shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/units/
5/test/items/7/weights
```

Body:

```
content-type: application/json
```

```
{
  "weight" : 4,
  "cid": 3
}
```

Nota: El campo "weight", corresponde al peso asignado al item con respecto a un concepto este valor debe ser un entero entre 0 y 5. El campo "cid" corresponde al id del concepto a modificar el peso.

DELETE

```
-/instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/
items/{itemId}/weights
```

Ejemplo:

```
URI:shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/
units/5/test/items/7/weights
```

Body:

```
content-type: application/json
```

```
{
  "weight" : 0,
  "cid": 3
}
```

Nota: El campo "weight", corresponde al peso asignado al item con respecto a un concepto en este caso puede llevar cualquier valor entero. El campo "cid" corresponde al id del concepto al que se le asignó el peso.

LOGIN:

GET

```
-/auth
```

Ejemplo:

```
URI: shirepo/webapi/auth
Body:
content-type: application/json
{
  "userEmail" : "juan.ramirezpadilla@gmail.com",
    "password" : "password",
    "userType": 1
}
```

Nota: El campo "userEmail", corresponde al email del usuario. El campo "password", corresponde a la contraseña del usuario. El campo "userType", corresponde al tipo de usuario: 0 si es estudiante o 1 si es instructor.

RECURSOS:

GET

```
-/instructors/{email}/courses/{courseId}/structure/units/{unitId}/
concepts/{conceptId}/resources
```

POST

```
-/instructors/{email}/courses/{courseId}/structure/units/{unitId}/
concepts/{conceptId}/resources
```

Ejemplo:

```
URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/resources
```

Body:

```
content-type: application/json
{
  "text" : "La P00 es una técnica para desarrollar soluciones computacionales
utilizando componentes de software (objetos de software). Objeto: Componente o
código de software que contiene en sí mismo tanto sus características
(campos) como sus comportamientos (métodos); se accede a través de su interfaz
o signatura."
}
```

Nota: El campo "text", corresponde a la explicación de un concepto.

PUT

```
-/instructors/{email}/courses/{courseId}/structure/units/{unitId}/concepts/{conceptId}
```

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/resources/1

Body:

content-type: application/json

```
{
  "text" : "La P00 es una técnica para desarrollar soluciones computacionales
  utilizando componentes de software (objetos de software). Objeto: Componente o
  código de software que contiene en sí mismo tanto sus características
  (campos) como sus comportamientos (métodos); se accede a través de su interfaz
  o signatura."
}
```

Nota: El campo "text", corresponde a la explicación de un concepto.

DELETE

```
-/instructors/{email}/courses/{courseId}/structure/units/{unitId}/
concepts/{conceptId}/resources/{resourceId}
```

RESPUESTAS:

GET

```
- /instructors/{email}/courses/{courseId}/structure/units/{unitId}/
test/items/{itemId}/answers
```

POST

```
- /instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/
items/{itemId}/answers
```

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/units/7/test/items/9/answers

Body:

content-type: application/json

```
{
  "text" : "Es nivel más alto de protección de datos",
  "correct" : false
}
```

Nota: El campo "text", corresponde al enunciado de la respuesta. El campo "correct", indica si la respuesta es correcta ("true") o errónea ("false").

PUT

```
- /instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/items/{itemId}/answers/{answerId}
```

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/units/7/test/items/9/answers/12

Body:

content-type: application/json

```
{
  "text" : "Es nivel más alto de protección de datos conocido",
  "correct" : false
}
```

Nota: El campo "text", corresponde al enunciado de la respuesta. El campo "correct", indica si la respuesta es correcta ("true") o errónea ("false").

DELETE

```
- /instructors/{email}/courses/{courseId}/structure/units/{unitId}/test/items/{itemId}/answers/{answerId}
```

SESIONES:

GET

```
- /courses/{courseId}/session/all
- /course/{courseId}/session/{sessionId}
- /instructors/{email}/courses/{courseId}/session
```

PUT

```
- /instructors/{email}/courses/{courseId}/session
```

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/session/

Body:

content-type: application/json

```
{
  "id": 3,
  "status": true
}
```

Nota: Aquí se actualiza el estado de la sesión a "true" indicando que la sesión del curso está activa, por lo que el curso aparecerá en la lista de cursos disponibles.

UNIDADES:**GET**

- /courses/{courseId}/structure/units/all
- /instructors/{email}/courses/{courseId}/structure/units

POST

- /instructors/{email}/courses/{courseId}/structure/units

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/units

Body:

```
contenttype: application/json
{
  "nodeName" : "Encapsulamiento",
  "turn" : 1
}
```

Nota: El campo "nodeName", que corresponde al nombre de la unidad. El campo "turn", corresponde a la secuencia/ubicación del nodo unidad en la estructura (árbol) del curso.

PUT

- /instructors/{email}/courses/{coursesId}/structure/units/{unitId}

Ejemplo:

URI: shirepo/webapi/instructors/juan.ramirezpadilla/courses/2/structure/units/7

Body:

```
content-type: application/json
{
  "nodeName" : "Clases",
  "turn" : 1
}
```

Nota: El campo "nodeName", que corresponde al nombre de la unidad. El campo "turn", corresponde a la secuencia/ubicación del nodounidad en la estructura (árbol) del curso. El parámetro "unitId" en la URI, corresponde al id de una unidad existente en la base de datos.

DELETE

- /instructors/{email}/courses/{coursesId}/structure/units/{unitId}

DIAGNÓSTICO:**GET**

`-/students/{email}/diagnostic?unitId=:value`

Ejemplo:

URI: `shirepo/webapi/instructors/students/jorgeramirezp96/diagnostic?unitId=2`

Nota: El campo "unitId", corresponde al identificador de la unidad que está siendo evaluada.

POST

`-/students/{email}/diagnostic`

Ejemplo:

URI: `shirepo/webapi/instructors/students/jorgeramirezp96/diagnostic`

Body:

`content-type: application/json`

```
[{"courseId":1,"unitId":2,"itemId":3,"answerId":4},
{"courseId":1,"unitId":2,"itemId":5,"answerId":1}]
```

Nota: Es necesario enviar un array con las respuestas dadas por el estudiante. El campo "courseId", corresponde al identificador del curso que está realizando el estudiante. El campo "unitId", corresponde al identificador de la unidad que está siendo evaluada. El campo "itemId", corresponde al identificador del item (pregunta) que contestó el estudiante. El campo "answerId", corresponde al identificador de la respuesta que seleccionó el estudiante.

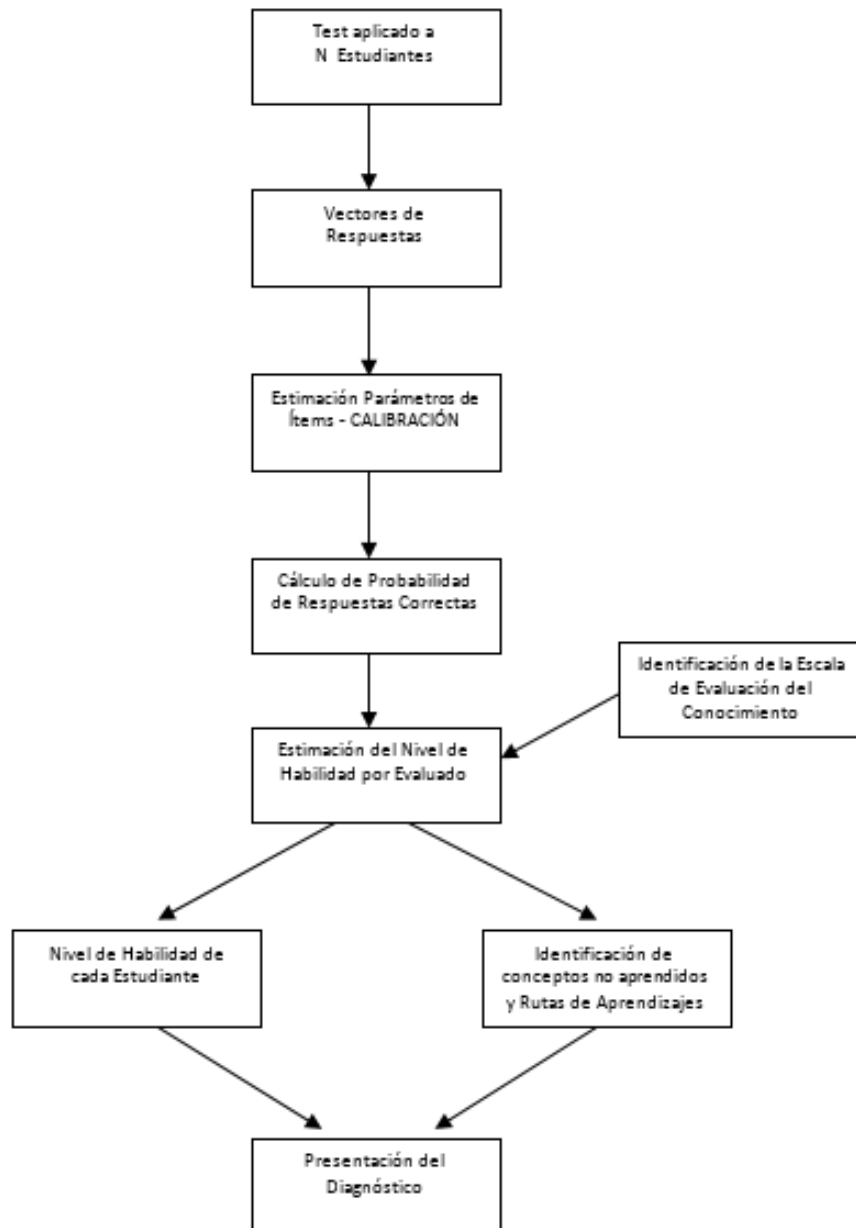


Figura 4-11: Funcionamiento General del Modelo

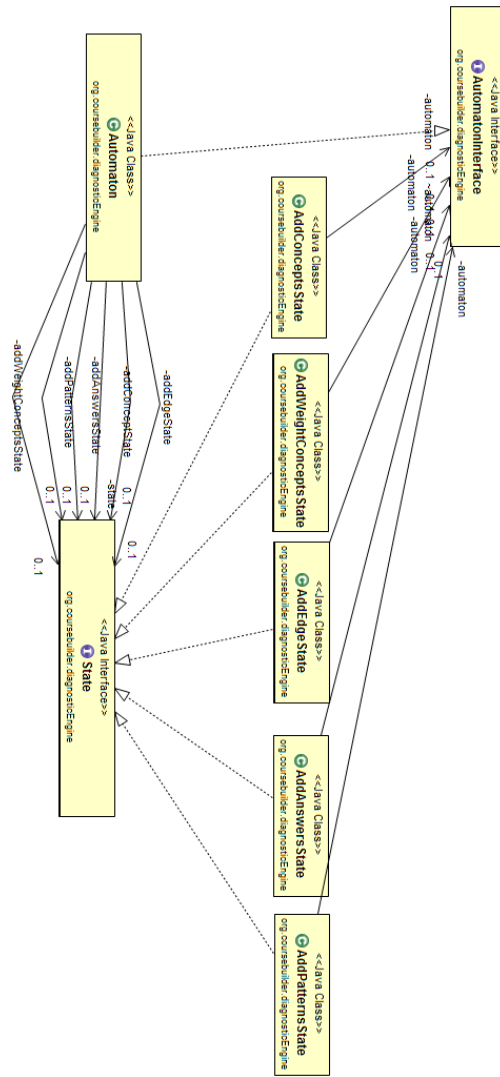


Figura 4-12: Diagrama de Clases DIAGEN

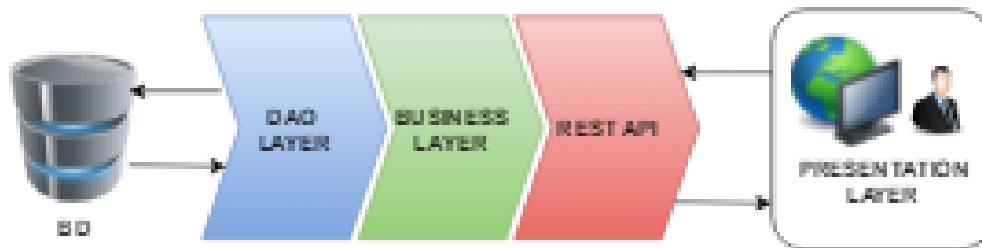


Figura 4-13: Arquitectura Aplicación Web

5 Pruebas y Validación

5.1. Introducción

Una vez elaborados los recursos que componen el servicio web con las funcionalidades del LMS, se procede a elaborar una aplicación web modelo que servirá de referencia al momento de consumir el servicio web elaborado y para validar la solución propuesta. A continuación se detalla la elaboración de dicha aplicación.

5.2. Aplicación Web Modelo

Esta aplicación propicia la interacción entre el usuario con el sistema, con el fin de validar la funcionalidad del servicio web implementado. Se agruparán las interfaces en función del tipo de usuario que interactúa con la aplicación. El sistema presenta 2 tipos de usuarios:

A Instructor: Este tipo de usuario tiene la capacidad de crear y consultar cursos propios, describir unidades y conceptos de cada curso, proporcionar el material relacionado a estos y diseñar evaluaciones. Toda esta información será consumida por los usuarios Estudiantes. Para lo anterior, el sistema ofrece las siguientes interfaces:

- a Cursos: En esta interfaz, el Instructor puede iniciar el proceso de creación de un nuevo curso. Adicionalmente, puede consultar los cursos que haya creado previamente. La interfaz ofrece las siguientes opciones:

Creación de Cursos. Esta opción inicia con un proceso que se divide en 3 pasos:

- Información del Curso: El Instructor proporcionará datos del curso como nombre, descripción, duración, etiquetas (tags) de la temática del curso, que serán suficientes para identificar el curso. Ver Figura 5-1.
 - Estructura del Curso: En este paso, el Instructor ingresará el contenido temático del curso (ítems) según el modelo detallado en la sección 4.3.1.
 - Material de estudio: En este paso, el Instructor ingresará, por cada concepto ingresado en el paso anterior, todo el material de estudio (texto o audiovisual) necesario para dominar dicho concepto.

Consulta de Cursos. Al momento del ingreso a la interfaz de cursos, se presentan los cursos que el Instructor ha creado previamente. Ver Figura 5-2. Cada curso

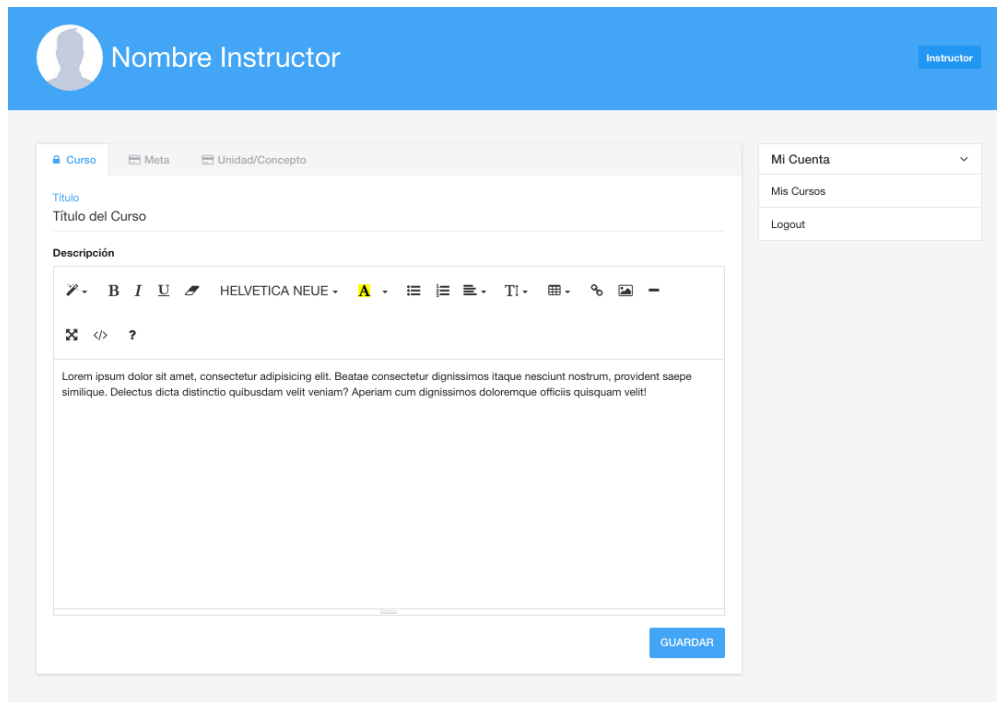


Figura 5-1: Edición del Curso por Instructor

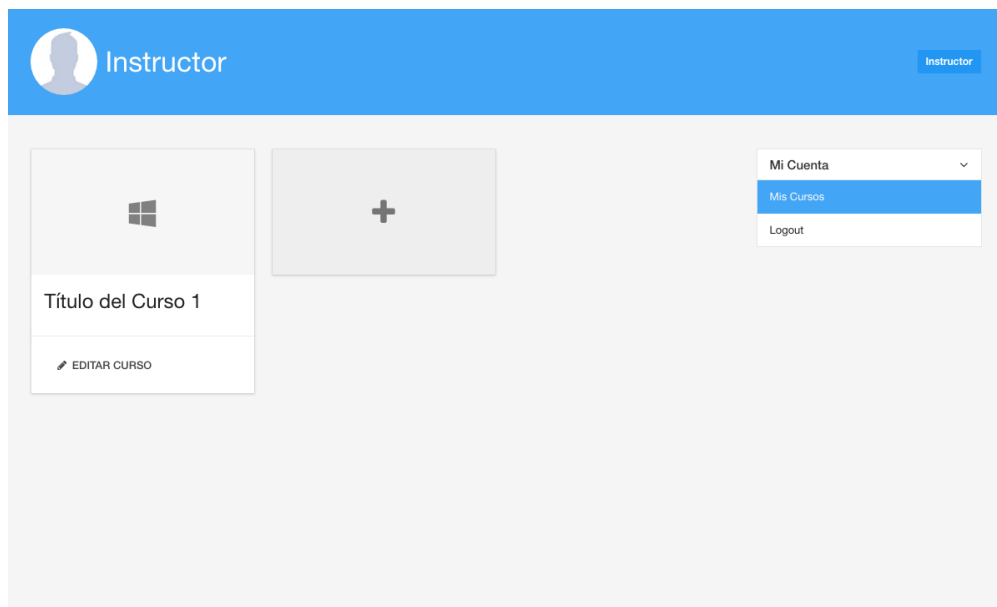


Figura 5-2: Listado de Cursos por Instructor

listado, debe tener un enlace que dirija al Instructor a una vista detallada del Curso, en donde se podrán realizar los siguientes procesos:

- Edición del Curso: El Instructor tendrá acceso nuevamente a la interfaz de Creación de Cursos, con los datos del curso cargados. Estos datos podrán modificarse según se requiera.
- Inactivación del Curso: El instructor podrá inhabilitar u ocultar un curso, según se requiera.
- Eliminación del Curso: El instructor podrá eliminar un curso, según se requiera.

b Evaluaciones: En esta interfaz, el Instructor puede crear una evaluación, en función de la estructura de un curso previamente creado:

Crear evaluación. El proceso de creación de una evaluación se divide en N pasos:

- Selección del Curso: El Instructor elegirá el curso cuyo modelo de conceptos será sujeto a evaluación.
- Creación de preguntas y respuestas: El Instructor podrá construir preguntas y respuestas para la evaluación. Para cada pregunta, el Instructor ingresa el enunciado, y puede crear tantas respuestas desee. Para cada respuesta, debe ingresar el texto correspondiente y un estado de precisión (Correcta o Incorrecta). Posterior a esto se escogen, desde el modelo de conceptos del curso escogido, los conceptos que serán evaluados por la pregunta, siguiendo la metodología descrita en la sección 4.3.3. Ver Figura 5-3.

B Estudiante: Este tipo de usuario tiene la capacidad de consumir los cursos creados por los Instructores. El Estudiante podrá consultar y suscribirse a cursos. Al realizar los cursos, eventualmente consumirá las evaluaciones creadas por los Instructores para cada curso. Para lo anterior, el sistema ofrece las siguientes interfaces:

Cursos. En esta interfaz, el Estudiante puede iniciar el proceso de suscripción a un curso. También puede consultar los cursos a los que esté suscrito, y retomar la actividad de estos. La interfaz ofrece las siguientes opciones:

Detalle de Curso. Cada curso listado en la interfaz Cursos, debe tener un enlace que lleve al Estudiante a esta sección. Una vez aquí, el usuario puede conocer en detalle la información del curso (nombre, descripción, duración, contenido temático, etc. El contenido temático será una representación del modelo de conceptos detallado en la sección 4.3.1). Ver Figura 5-4. El Estudiante podrá realizar las siguientes operaciones:

- Suscripción al curso.
- Retomar curso: En este paso, el Estudiante puede ingresar al módulo de desarrollo del curso.

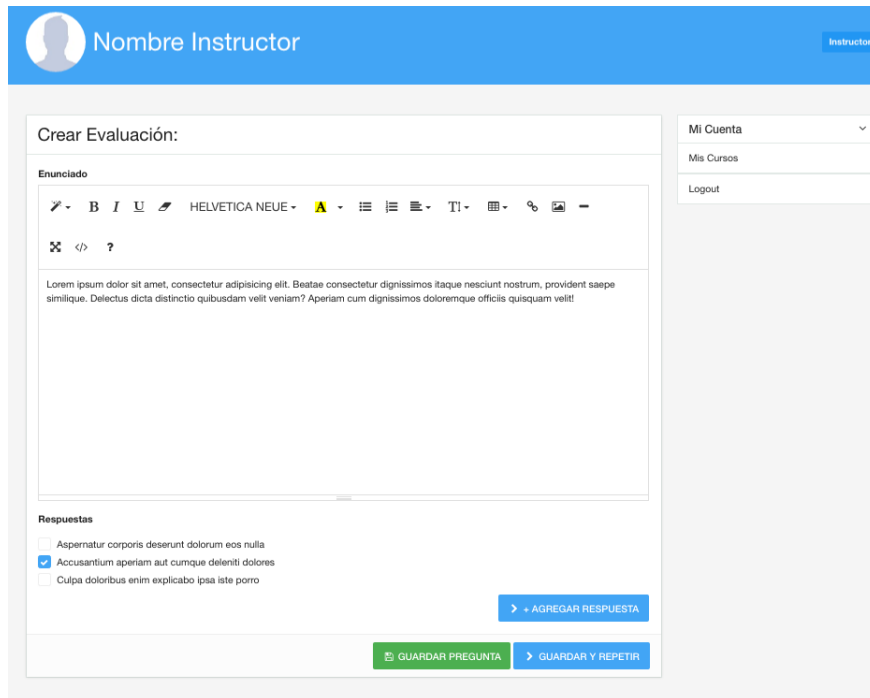


Figura 5-3: Creación de Evaluación por Instructor

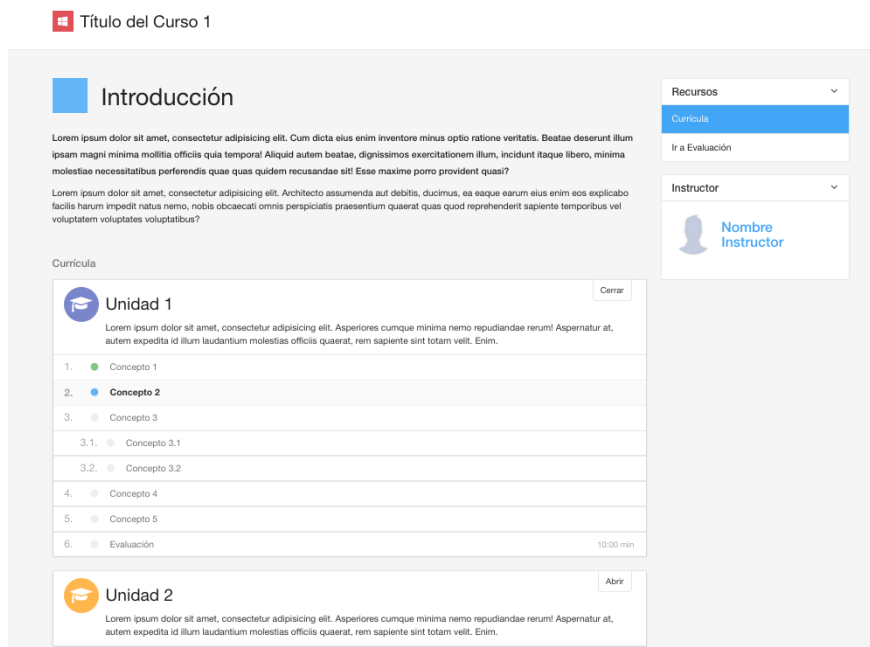


Figura 5-4: Vista Curso de Estudiante

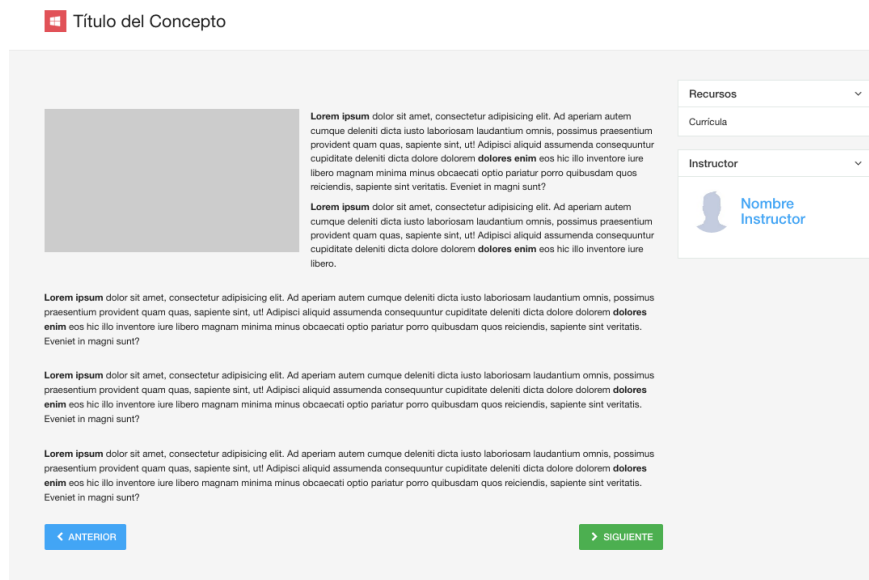


Figura 5-5: Vista de Materiales de Estudiante

Módulo de desarrollo del Curso. En esta interfaz, el Estudiante tiene acceso al contenido temático del curso (representación del modelo de conceptos), evaluaciones, y podrá retirarse del curso en cualquier momento. El Estudiante podrá realizar las siguientes operaciones:

- Acceso al material/recurso de estudio: El cada ítem (concepto) del contenido temático, debe tener un enlace al material de estudio registrado para dicho ítem (concepto). Adicionalmente, se tiene un enlace hacia el siguiente material de estudio, en función del concepto actual. Ver Figura 5-5.
- Acceso a evaluaciones: Una vez que el estudiante finaliza el consumo del material de estudio de los conceptos de una Unidad, el Estudiante tendrá acceso a la evaluación creada para dicha Unidad o conceptos de dicha Unidad.

Evaluaciones. En esta interfaz, el Estudiante tiene acceso a las evaluaciones que tiene disponible, en función de los conceptos/Unidades que haya consumido. Al seleccionar una evaluación, el Estudiante inicia el siguiente proceso:

- Realizar evaluación: El Estudiante inicia la evaluación, la cual tendrá un límite de tiempo. Ver Figura 5-6.
- Retroalimentación: Al finalizar de la evaluación, el Estudiante obtendrá un reporte correspondiente a los conceptos que debe reforzar (de ser necesario) para tener un dominio suficiente de los conceptos evaluados. Este reporte se conoce como “camino de aprendizaje”, detallado en la sección 4.3.3.

Figura 5-6: Vista Evaluación de Estudiante



Figura 5-7: Estructura Jerárquica de la Temática

- Historial de evaluaciones: El Estudiante podrá consultar sus resultados históricos, junto con los caminos de aprendizaje obtenidos.

Para validar el funcionamiento de la herramienta, se contruyó un curso modelo de Programación Orientada a Objetos. Este curso consta de una unidad (Introducción a la programación orientada a objetos) en la que se crea una estructura jerárquica a partir de la temática tratada. La Figura 5-7 describe la estructura de la temática. Además, se elaboró un test con 5 ítems y se establecieron los pesos de las relaciones entre estos y los conceptos (Ver Figura 5-8). Vale agregar que se considera que un concepto es débil (El estudiante presenta falencias) si obtiene un puntaje menor a 0.6 una vez realizado el diagnóstico.

Para evaluar el recurso de diagnóstico de falencias conceptuales de la implementación, se simuló 3 casos de prueba:

	Polimorfismo	Encapsulación	Herencia	Clase	Objeto	POO
I1	0	0	0	5	3	1
I2	0	0	5	4	0	0
I3	0	0	0	2	5	0
I4	5	2	3	2	0	0
I5	0	5	0	0	3	0

Figura 5-8: Tabla Ítem vs Conceptos

	Polimorfismo	Encapsulación	Herencia	Clase	Objeto	POO
I1	0	0	0	5	3	1
I2	0	0	5	4	0	0
I3	0	0	0	2	5	0
I4	5	2	3	2	0	0
I5	0	5	0	0	3	0
TP CI	5	7	8	13	11	1
Ri	0	0	0	0	0	0
PRc	1	1	1	1	1	1

Figura 5-9: Tabla Ítem vs Conceptos Caso 1

1. Estudiante acierta todas los ítems.

La Figura 5-9 muestra la tabla ítem vs conceptos para este caso. El estudiante obtiene un valor de PRc de 1 para todos sus conceptos. Esto indica que ningún concepto es débil debido a que sus puntajes son mayor a 0.6 (Ver Figura 5-10). Por consiguiente no se presentan caminos de aprendizaje (Ver Figura 5-11).

2. Estudiante falla 1 ítem. La Figura 5-12 muestra la tabla ítem vs conceptos para este caso. El estudiante falla el ítem 4 y obtiene un valor de PRc de 0, 0.71, 0.63, 0.85, 1, 1 para los conceptos Polimorfismo, Encapsulación, herencia, Clase, Objeto y POO, respectivamente. Esto indica que el concepto Polimorfismo es el único concepto débil debido a que su puntaje es menor a 0.6 (Ver Figura 5-13). En otras palabras el estudiante presenta falencias conceptuales en este concepto y en el camino de aprendizaje sólo se sugiere estudiar Polimorfismo (Ver Figura 5-14).

3. Estudiante falla todos los ítems. La Figura 5-15 muestra la tabla ítem vs conceptos para

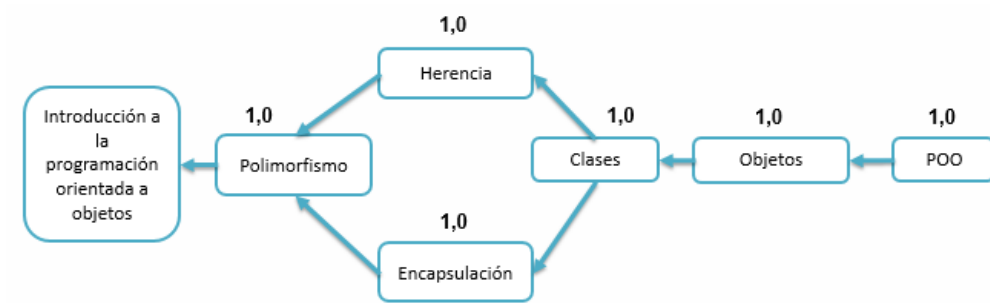


Figura 5-10: PRc Conceptos Caso 1

Resultado

CAMINOS DE APRENDIZAJE:

::

Puntuación: 100%

Figura 5-11: Diagnóstico Caso 1

	Polimorfismo	Encapsulación	Herencia	Clase	Objeto	POO
I1	0	0	0	5	3	1
I2	0	0	5	4	0	0
I3	0	0	0	2	5	0
I4	5	2	3	2	0	0
I5	0	5	0	0	3	0
TP CI	5	7	8	13	11	1
Ri	5	2	3	2	0	0
PRc	0	0.71	0.63	0.85	1	1

Figura 5-12: Tabla Ítem vs Conceptos Caso 2



Figura 5-13: PRc Conceptos Caso 2

Resultado
CAMINOS DE APRENDIZAJE:
::
:: Polimorfismo
Puntuación: 80%

Figura 5-14: Diagnóstico Caso 2

este caso. El estudiante obtiene un valor de PRc de 0 para todos sus conceptos. Esto indica que todos los conceptos son débiles debido a que sus puntajes son menor a 0.6 (Ver Figura 5-16). En otras palabras el estudiante presenta falencias conceptuales en todos los conceptos y se generan caminos de aprendizaje para todos ellos(Ver Figura 5-17).

	Polimorfismo	Encapsulación	Herencia	Clase	Objeto	POO
I1	0	0	0	5	3	1
I2	0	0	5	4	0	0
I3	0	0	0	2	5	0
I4	5	2	3	2	0	0
I5	0	5	0	0	3	0
TP CI	5	7	8	13	11	1
Ri	5	7	3	13	11	1
PRc	0	0	0	0	0	0

Figura 5-15: Tabla Ítem vs Conceptos Caso 3

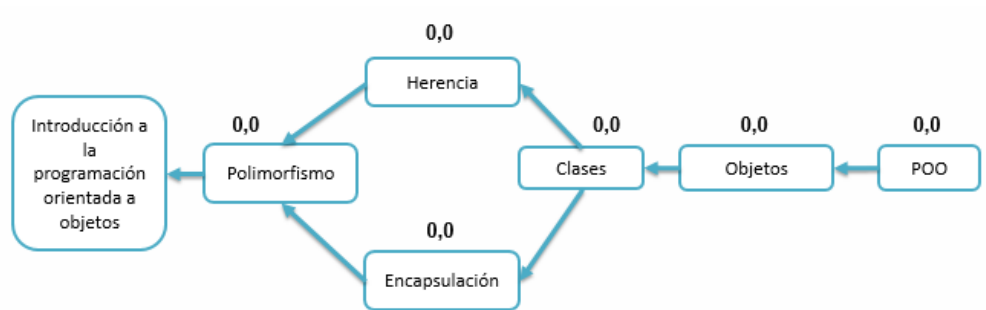


Figura 5-16: PRc Conceptos Caso 3

Resultado

CAMINOS DE APRENDIZAJE:

- ::
- :: Polimorfismo
- :: Encapsulación -> Polimorfismo
- :: Herencia -> Polimorfismo
- :: Clase
- :: Clase -> Encapsulación Polimorfismo
- :: Clase -> Herencia -> Polimorfismo
- :: Objeto -> Clase
- :: POO -> Objeto -> Clase

Puntuación: 0%

Figura 5-17: Diagnóstico Caso 3

5.3. Conclusiones

La aplicación web modelo presenta funcionalidades básicas de un LMS, como lo son: Creación y consumo de cursos académicos, control y evaluación del conocimiento de conceptos por parte del estudiante. Esta aplicación sirve como punto de referencia para implementaciones futuras.

El servicio web posee los recursos necesarios para cubrir las funcionalidades antes mencionadas. A su vez, dota de flexibilidad e interoperabilidad al sistema, permitiendo que se implementen aplicaciones web bajo criterios y necesidades diferentes, y dada su naturaleza RESTFUL, el servicio web es independiente de la plataforma y lenguajes de programación usados en su implementación.

6 Conclusiones y Trabajos Futuros

6.1. Conclusiones

Se elaboró un comparativo de LMS con el que se identificaron sus funcionalidades, ventajas y desventajas. El LMS elaborado en este trabajo cumple con los estándares de diseño y con las funcionalidades básicas con las que debe contar.

Para la elaboración de los cursos se identificaron las características a través de un comparativo. A partir de lo anterior se elaboró una metodología para el diseño de los cursos que conforman el LMS desarrollado.

Se identificó que la arquitectura orientada a capas que propone JEE es la que más se ajusta al desarrollo del LMS propuesto. El diseño y elaboración de los módulos se hicieron con base en esta arquitectura la cual permite que la solución sea modular, flexible y escalable.

La capa de persistencia y de acceso a los datos fue elaborada con Hibernate, lo cual proporciona la posibilidad de trabajar con múltiples sistemas gestores de bases de datos y la opción de construir automáticamente tablas a partir de clases. La capa de lógica de negocios y servicios fue elaborada con Spring y Jersey, los cuales nos brindan por una parte inyección de dependencias y por otra un conjunto de herramientas para elaborar servicios web RESTFUL. En esta capa se encuentran el conjunto de servicios elaborados, en los que se destaca el servicio de diagnóstico de estudiantes, que permite identificar un conjunto de conceptos en los que el estudiante está presentando cierta dificultad en el aprendizaje. Los servicios permiten al desarrollador construir su LMS a partir de las funcionalidades que necesite.

La aplicación web elaborada para consumir la API RESTFUL, puede ser elaborada según las necesidades de su implementación, la capa de servicios web permite que la elaboración de esta capa sea flexible. Una vez puesto en funcionamiento una aplicación que consuma la API REST los profesores podrán elaborar tanto cursos secuenciales como no secuenciales. Además, los tests ofrecen una evaluación específica de cada concepto, lo que a su vez permiten recibir un diagnóstico automático de su proceso de aprendizaje, por lo que no se hace necesaria la intervención del profesor.

El LMS elaborado incrementa la eficiencia y efectividad del aprendizaje asistido por compu-

tadora (e-learning) en terminos de reusabilidad, interoperabilidad, accesibilidad y modularización.

Por último, la funcionalidad de generar diagnósticos del LMS orientado a servicios desarrollado en este trabajo, le permite al evaluado evidenciar en qué conceptos tiene debilidades, lo que facilitará su proceso de refuerzo y autoaprendizaje.

6.2. Trabajo Futuro

Como trabajos futuros se proponen:

- La evolución del LMS a un Sistema Tutor Inteligente, esto con el fin de elaborar caminos de aprendizajes óptimos para cada estudiante de manera automática. Se recomienda expandir el conjunto de servicios web, debido a que los elaborados en el actual trabajo, a excepción del servicio de diagnóstico, realizan las funciones básicas que todo LMS debe tener.
- La generación de Test Adaptativos Informatizados (TAI) que pueden ser aplicados al evaluado después de haber obtenido el diagnóstico inicial, en el cual se han identificado los conceptos en los que presenta falencias.
- Diseñar un aplicativo que permita identificar que evaluados comparten las mismas debilidades conceptuales (Generación de clusters por similaridad de conceptos débiles).
- Extender el servicio web para hacer posible la adición de diversos recursos educativos, que faciliten el proceso de aprendizaje del estudiante.

Bibliografía

- [Alonso et al., 2004] Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2004). Web services: Concepts, architecture and applications. *Springer Verlag*.
- [Berners-Lee, 1996] Berners-Lee, T. (1996). Hypertext transfer protocol - http/1.0. ietf rfc.
- [Bianco et al., 2007] Bianco, P., Kotermanski, R., and Mersion, P. (2007). Evaluating a service-oriented architecture.
- [Brusilovsky, 1999] Brusilovsky, P. (1999). Adaptive and intelligent technology for web-based education. *Special Issue on Intelligent Systems and Teleteaching*, 4:19–25.
- [Brusilovsky and Weber, 2001] Brusilovsky, P. and Weber, G. (2001). Elm-art: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12:351–384.
- [Christensen, 2001] Christensen, E. (2001). Web services description language (wsdl) 1.1. w3c note. w3c.
- [Chu et al., 2006] Chu, H. C., Hwang, G. J., Tseng, J. C., and Hwang, G. H. (2006). A computerized approach to diagnosing student learning problems in health education. *Asian Journal of Health and Information Sciences*, 1:43–60.
- [Conejo et al., 2001] Conejo, R., Millán, E., Perez-de-la Cruz, J. L., and Trella, M. (2001). Modelado del alumno: Un enfoque bayesiano. *Revista Iberoamericana de Inteligencia Artificial*, 12:50–58.
- [Coursera, 2016] Coursera (2016). Url: <https://coursera.org>.
- [Cronbach, 1951] Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, pages 297–334.
- [Díaz et al., 1996] Díaz, P., Catenazzi, N., and Aedo (1996). De la multimedia a la hipermedia.
- [Embreston, 1985] Embreston, S. E. (1985). Introduction to the problem of test design. *Susan. E. Embreston: Test Design: Developments in psychology and psychometrics. Orlando, Fl. EE. UU. Academic Press. INC.*

- [Fielding, 1997] Fielding, R. (1997). Hypertext transfer protocol – http/1.1. ietf rfc 2068.
- [Fielding, 1999] Fielding, R. (1999). Hypertext transfer protocol – http/1.1. ietf rfc 2616.
- [Fielding, 2004] Fielding, R. (2004). Rfc for rest. rest discussion mailing list.
- [Fielding, 2000] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Tesis de doctorado, Universidad de California.
- [Fragoso et al., 2014] Fragoso, O. G., Santaolaya, R., Muñoz, S. J., Valenzuela, B. D., and Rojas, J. C. (2014). Integration of learning web services into learning management systems. *Central America and Panama Convention*, pages 1–6.
- [FROAC, 2016] FROAC (2016). Federación de repositorios de objetos de aprendizaje froac url: <http://froac.manizales.unal.edu.co/roap/main.php>.
- [Gudgin, 2007] Gudgin, M. (2007). Soap version 1.2 part 1: Messaging framework (second edition). w3c recommendation.
- [Hambleton et al., 1991] Hambleton, R. K., Swaminathan, J., and Rogers, H. J. (1991). Fundamentals of item response theory. *Sage publications*.
- [Hsu and Gupta, 1998] Hsu, Y. C. and Gupta, S. K. (1998). A new path-oriented effect-cause methodology to diagnose delay failures. *Test Conference*, pages 758–767.
- [Hwang, 2003] Hwang, G. J. (2003). A concept map model for developing intelligent tutoring systems. *Computers and amp*, pages 217–235.
- [Hwang, 2005] Hwang, G. J. (2005). International journal of distance education technology. *Journal of Information Science and Engineering*, 3(4):35–50.
- [Hwang, 2007] Hwang, G. J. (2007). Gray forecast approach for developing distance learning and diagnostic systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37.
- [Hwang and Hsiao, 2003] Hwang, G. J. and Hsiao, J. L. (2003). A computer-assisted approach for diagnosing student learning problems in engineering courses. *Journal of Information Science and Engineering*, 19(2):229–248.
- [Jiménez and Ovalle, 2012] Jiménez, M. and Ovalle, M. (2012). Modelo de evaluación adaptativa del nivel de conocimiento del estudiante para sistemas tutoriales inteligentes. Master’s thesis, UNIVERSIDAD NACIONAL DE COLOMBIA - SEDE MEDELLÍN.
- [Kobsa and Wahlster, 1989] Kobsa, A. and Wahlster, W. (1989). User modelling in dialog systems. *Springer-Verlag*.

- [Lord, 1952] Lord, F. M. (1952). A theory of test scores. *Psychometric Monographs*.
- [Lord and Novick, 1968] Lord, F. M. and Novick, M. R. (1968). Statistical theories of mental test scores. *New York: Addison-Wesley*.
- [Moodle, 2016] Moodle (2016). Url: <https://moodle.org>.
- [Muñiz Fernández, 1997] Muñiz Fernández, J. (1997). Introducción a la teoría de respuesta a los ítems. *Madrid: Pirámide*.
- [Muñiz Fernández, 2010] Muñiz Fernández, J. (2010). Las teorías de los test: Teoría clásica y teoría de respuestas a los Ítems. *Redalyc*.
- [MVA, 2016] MVA (2016). Microsoft virtual academy url: <https://mva.microsoft.com>.
- [Navas, 1994] Navas, M. J. (1994). Teoría clásica de los tests versus teoría de respuesta al ítem. *Psicológica*, 15:175–208.
- [Nelson, 1972] Nelson, T. H. (1972). From memex to hypertext: Vannevar bush and the mind's machine.
- [Nielsen, 1990] Nielsen, J. (1990). Communications of the acm. *Springer-Verlag*, 33.
- [Nielsen, 1998] Nielsen, J. (1998). Usability engineering. *Academic Press: Londres (Gran Bretaña)*.
- [Oliver, 1995] Oliver, R. (1995). Interactive information systems: Information access and retrieval. *Electronic Library*, 33:187–193.
- [Ortega Molina, 2005] Ortega Molina, F. D. (2005). *Sistemas hipermedia para el aprendizaje de la Lectoescritura Sistemas Hipermedia para el aprendizaje de la Lectoescritura*. Tesis de doctorado, Universidad de Cádiz.
- [Partheeban and SankarRam, 2014] Partheeban, N. and SankarRam, N. (2014). e-learning management system using web services. *Information Communication and Embedded Systems*, pages 1–7.
- [Pautasso et al., 2008] Pautasso, C., Zimmermann, O., and Leymann, F. (2008). Restful web services vs. "big" web services: Making the right architectural decision.
- [Phankokkrud and Woraratpanya, 2009] Phankokkrud, M. and Woraratpanya, K. (2009). Web services for learning management systems: Communication architecture. *2009 IEEE 9th Malaysia International Conference*, pages 403–408.
- [Ramaratnam, 2007] Ramaratnam, R. (2007). An analysis of service oriented architectures.

- [Repositorio de Objetos de Aprendizaje, 2016] Repositorio de Objetos de Aprendizaje, R.-U. (2016). Url: <http://www.rad.unam.mx/>.
- [Richardson and Ruby, 2007] Richardson, L. and Ruby, S. (2007). Restful web services.
- [Robles Pedrozo and Rodriguez-Artacho, 2012] Robles Pedrozo, L. S. and Rodriguez-Artacho, M. (2012). Descubrimiento de problemas de aprendizaje a través de test: Fiabilidad y metodología de diagnóstico basado en clustering. Master's thesis, UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA - UNED.
- [Romero, 2005] Romero, F. (2005). Sistemas hipermedia en la enseñanza : Elementos de análisis y tradiciones de reflexión panorámica histórica definición y conceptos vinculados posibilidades argumentos en contra texto vs . hipermedia. page 1–50.
- [Rouet, 2000] Rouet, J. (2000). Hypermedia and learning- cognitive perspectives. *Journal of Computer Assisted Learning*, 16:97–101.
- [Santisteban and Alvarado, 2001] Santisteban, C. and Alvarado, J. (2001). Modelos psicométricos. *Madrid: UNED*.
- [Sharma and Vatta, 2013] Sharma, A. and Vatta, S. (2013). Role of learning management systems in education. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3:997–1002.
- [Su et al., 2007] Su, M. T., Wong, C. S., Soo, C. F., Ooi, C. T., and Sow, S. L. (2007). Service-oriented e-learning system. *Information Technologies and Applications in Education*, pages 6–11.
- [Thurstone, 1925] Thurstone, L. L. (1925). A method of scaling psychological and educational tests. *Journal of Educational Psychology*, 16:433–451.