

Universidad de las Ciencias Informáticas

Facultad 2



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

Título: Modelo del profesor como parte de un Sistema Tutorial Inteligente para recomendar de forma diferenciada la bibliografía adecuada a los estudiantes en la asignatura Redes y Seguridad Informática.

Autores:

Alejandro Mojena Elías
José Alejandro Perez Rodriguez

Tutor:

Ing. Iralys Torres Pérez

La Habana.
Junio, 2015.



El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia (...)"

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los ___ del mes de Junio de 2015.

Alejandro Mojena Elías

José Alejandro Perez Rodriguez

Autor

Autor

Ing. Iralys Torres Pérez

Tutor

AGRADECIMIENTOS

Alejandro

A mis padres mil gracias por apoyarme siempre, hoy soy lo que soy gracias a ellos. Por siempre confiar en mí y darme fuerzas cuando más lo necesité, gracias mami por decirme siempre que todo iba a salir bien. Por todo el amor que me han dado y por todo lo que me han enseñado. Los amo.

A mi hermano por ser lo que más quiero en la vida, por todo su cariño, ojalá y algún día él pueda pasar por esta experiencia tan bella.

A mis abuelos, por ser más que abuelos madres y padres, gracias por quererme tanto y querer complacerme en todo.

A mis tíos, por el amor que me han brindado siempre y todos los consejos que me han dado.

A todos mis primos, por quererme tanto, en especial a Isabela por ser tan adorable conmigo.

A toda mi familia en general por siempre apoyarme, los quiero.

A mi novia, que también es parte de mi familia, gracias por el amor que me brindas, por querer complacerme en todo y por ser la mujer que más he amado en la vida. Te amo mi reina.

AGRADECIMIENTOS

José Alejandro

A mis padres que son los mejores del mundo, que son los que me han formado con mucho cariño estando siempre en todos los momentos importantes de mi vida y dándome su incondicional apoyo, son incomparables y siempre han sido mi mayor orgullo.

A mi hermana por ser mi mayor ejemplo y mi guía a seguir, por apoyarme muchísimo a lo largo de mis estudios y por estar a mi lado en el primer año de la carrera.

A mis abuelos en especial a mi abuelita caruca, por darme tanto cariño y amor, por cada día darme fuerza y apoyo, por enseñarme a querer la patria y por encima de todo a la hermosa familia que tengo, te adoro abuela.

A mi novia que ha estado a mi lado en estos cinco años que llevamos juntos, ayudándome en todo momento, sirviéndome de ejemplo como estudiante y por ser el amor de mi vida.

A mi hermano Ricardo, a todos mis primos y tíos.

A todos los profesores que desde edades tempranas me han ayudado, empezando por mi mamá y terminando por los de la UCI, en especial a Madelís la madrina del grupo 2202 gracias por tu gran apoyo, a mí tutora Iralis y Yuliet por ser mi primera profesora de programación y mi otra tutora a todos los profes del tribunal por sus críticas constructivas.

Y de todo corazón a mis compañeros de batalla que me han soportado estos cinco años de vida, que ya son mi otra familia en especial a mi compañero de tesis por

elegirme para concretar esta gran tarea, sabes que puedes contar conmigo para lo que sea, eres una gran persona.

A todos muchísimas gracias por su apoyo.

DEDICATORIA

Alejandro

Le dedico mi tesis a mi hermano por ser la persona más importante en mi vida, deseando que algún día pueda pasar por esta experiencia y logre convertirse en un verdadero profesional.

José Alejandro

Dedico el siguiente trabajo a mis padres, a mi abuelita Caruca, a mi hermana y a mi novia, por ser las personas más importantes en mi vida.

RESUMEN

El avance de la ciencia y la tecnología ha permitido el desarrollo de aplicaciones informáticas en todas las esferas de la sociedad. En la educación se ha incrementado el uso de la tecnología en función de potenciar la calidad del proceso de enseñanza-aprendizaje, como vía esencial para la formación de las nuevas generaciones, posibilitando la transmisión de conocimientos de una forma más interactiva y diferenciada.

Actualmente la Universidad de las Ciencias Informáticas (UCI), se encuentra inmersa en la búsqueda de estrategias de enseñanza que permitan facilitar el proceso de formación de los estudiantes. Sin embargo, en esta universidad los estudiantes carecen de medios de enseñanza adaptables al nivel de conocimiento que presentan los mismos en la asignatura Redes y Seguridad Informática. Además, los profesores no están disponibles en todo momento para asesorar de manera diferenciada la auto-preparación de los estudiantes en los temas de la asignatura. Todo ello ha contribuido a que el porcentaje de aprobados no sea el esperado por los profesores.

El presente trabajo tiene como objetivo, desarrollar un Sistema Tutorial Inteligente para personalizar el proceso de enseñanza-aprendizaje de la asignatura Redes y Seguridad Informática, mediante la recomendación de la bibliografía más adecuada para la auto-preparación de los estudiantes.

Palabras clave: Proceso de enseñanza-aprendizaje, Redes y Seguridad Informática, Sistema Tutorial Inteligente.

ÍNDICE

INTRODUCCIÓN.....	13
CAPÍTULO I: FUNDAMENTOS TEÓRICOS QUE SUSTENTAN EL DESARROLLO DE LOS SISTEMAS TUTORIALES INTELIGENTES.....	18
1.1 INTRODUCCIÓN	18
1.2 PROCESO DE ENSEÑANZA-APRENDIZAJE	¡ERROR! MARCADOR NO DEFINIDO.
1.3 SISTEMAS TUTORIALES INTELIGENTES.....	18
1.3.1 <i>Inteligencia Artificial</i>	¡Error! Marcador no definido.
1.3.2 <i>Componentes fundamentales de un Sistema Tutorial Inteligente:</i>	18
1.3.3 <i>El modelado del estudiante</i>	¡Error! Marcador no definido.
1.4 REFERENTES SOBRE TSI EN EL MUNDO, CUBA Y EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS	21
1.5 HERRAMIENTAS PARA EL DESARROLLO DE UN STI	¡ERROR! MARCADOR NO DEFINIDO.
1.6 SISTEMAS BASADOS EN EL CONOCIMIENTO	22
1.6.1 <i>Ingeniería del conocimiento en el desarrollo de sistemas basados en el conocimiento</i>	23
1.6.2 <i>Sistemas basados en reglas</i>	24
1.6.3 <i>Sistemas basados en probabilidades</i>	25
1.6.4 <i>Sistemas expertos conexionistas o redes neuronales artificiales</i>	25
1.6.5 <i>Sistemas basados en casos</i>	25
1.7 METODOLOGÍA DEL DESARROLLO	30
1.8 LENGUAJE DE PROGRAMACIÓN DEL LADO DEL SERVIDOR	31
1.9 LENGUAJE DE PROGRAMACIÓN DEL LADO DEL CLIENTE	31
1.10 FRAMEWORK WEB.....	32
1.11 HERRAMIENTA DE MODELADO	33
1.12 SISTEMA GESTOR DE BASE DE DATOS.....	33
1.13 SERVIDOR WEB	35
1.14 ENTORNO DE DESARROLLO INTEGRADO.....	35
1.15 CONCLUSIONES PARCIALES	36
CAPÍTULO 2: CARACTERÍSTICAS DEL STI.....	37
2.1 INTRODUCCIÓN	37
2.2 IMPACTO SOCIAL DEL SISTEMA TUTORIAL INTELIGENTE DE APOYO AL PROCESO DE ENSEÑANZA-APRENDIZAJE DE LA ASIGNATURA REDES Y SEGURIDAD INFORMÁTICA.	37
2.3 PROPUESTA DEL SISTEMA	38
2.3.1 <i>La Base de Casos</i>	38

2.3.2	<i>Fase de recuperación</i>	39
2.3.3	<i>Fase de adaptación</i>	41
2.3.4	<i>Fase de almacenamiento</i>	42
2.4	PERSONAS RELACIONADAS CON EL SISTEMA	42
2.5	LISTA DE RESERVA DEL PRODUCTO	42
2.6	FASE DE EXPLORACIÓN	44
2.6.1	<i>Historias de usuario</i>	44
2.6.2	<i>Clasificación de las HU</i>	45
2.6.3	<i>Historias de Usuarios del STI</i>	45
2.7	FASE DE PLANIFICACIÓN	47
2.7.1	<i>Plan de Iteraciones</i>	47
2.8	PLAN DE ENTREGAS	47
2.9	CONCLUSIONES PARCIALES.....	48
CAPÍTULO 3: DISEÑO, IMPLEMENTACIÓN Y PRUEBA		49
3.1	INTRODUCCIÓN	49
3.2	PATRÓN DE LA ARQUITECTURA	49
3.3	PATRONES DE DISEÑO	54
3.4	TARJETAS CLASE-RESPONSABILIDAD-COLABORADOR.....	57
3.5	MODELO DE DATOS.....	60
3.6	DIAGRAMA DE DESPLIEGUE	60
3.7	TAREAS DE INGENIERÍA.....	61
3.8	PRUEBAS.....	65
3.9	CONCLUSIONES PARCIALES.....	73
CONCLUSIONES GENERALES		75
REFERENCIAS.....		¡ERROR! MARCADOR NO DEFINIDO.

ÍNDICE DE TABLAS

TABLA 1. PERSONAS RELACIONADAS CON EL SISTEMA.	42
TABLA 2. HISTORIA DE USUARIO: AUTENTICAR USUARIO.....	46
TABLA 3. HISTORIA DE USUARIO: MOSTRAR CUESTIONARIO.	46
TABLA 4. HISTORIA DE USUARIO: MOSTRAR MATERIALES EDUCATIVOS.....	47
TABLA 5. PLAN DE ITERACIONES.	47
TABLA 6. PLAN DE ENTREGAS.	48
TABLA 7. TAREA DE INGENIERÍA: COMPROBAR EXISTENCIA DEL USUARIO EN LDAP Y ROL QUE OCUPA EN LA BASE DE DATOS.	65
TABLA 8. TAREA DE INGENIERÍA: MOSTRAR SECCIÓN CORRESPONDIENTE Y GUARDAR DATOS DE LA BASE DE DATOS.....	65
TABLA 9: CASO DE PRUEBA PARA EL CAMINO BÁSICO 1.	70
TABLA 10: CASO DE PRUEBA PARA EL CAMINO BÁSICO 8.....	70
TABLA 11: CASO DE PRUEBA PARA LA HU: AUTENTICAR USUARIO.....	72
TABLA 12: CASO DE PRUEBA PARA LA HU: MOSTRAR CUESTIONARIO.	73
TABLA 13: CASO DE PRUEBA PARA LA HU: MOSTRAR NOTA Y RECOMENDACIONES.....	73

ÍNDICE DE FIGURAS

FIGURA 1: ARQUITECTURA DE UN STI	19
FIGURA 2: ARQUITECTURA MVC DEL SISTEMA.	49
FIGURA 3: CLASES REPOSITORIOS DE LA CAPA DE ACCESO A DATOS.	50
FIGURA 4: ENTIDADES DE LA CAPA DE ABSTRACCIÓN DE BASE DE DATOS.....	51
FIGURA 5: PLANTILLA BASE.....	52
FIGURA 6: VISTAS DEL STI.	53
FIGURA 7: CLASES CONTROLADORAS.	54
FIGURA 8: EJEMPLO DEL PATRÓN EXPERTO EN EL STI.	55
FIGURA 9: EJEMPLO DEL PATRÓN CREADOR EN EL STI.	55
FIGURA 10: EJEMPLO DEL PATRÓN BAJO ACOPLAMIENTO EN EL STI.	56
FIGURA 11: EJEMPLO DEL PATRÓN ALTA COHESIÓN EN EL STI.....	57
FIGURA 12: TARJETA CRC DE LA CLASE: CASOCONTROLLER.	58
FIGURA 13: TARJETA CRC DE LA CLASE: CUESTIONARIOCONTROLLER.....	58
FIGURA 14: TARJETA CRC DE LA CLASE: PREGUNTACONTROLLER.	59
FIGURA 15: TARJETA CRC DE LA CLASE: INCISOCONTROLLER.	59
FIGURA 16: MODELO DE DATOS.....	60
FIGURA 17: DIAGRAMA DE DESPLIEGUE.	61
FIGURA 18: RESULTADO DE LA TÉCNICA DE PRUEBA RUTA BÁSICA.	71

INTRODUCCIÓN

En las últimas décadas, en la esfera de la educación, se han desarrollado herramientas informáticas que favorecen el aprendizaje y el conocimiento humano, explotando todas las facilidades que brindan las tecnologías de punta al proceso de enseñanza-aprendizaje. Un ejemplo de esto, es el desarrollo de los Softwares Educativos como aplicación informática de apoyo a la labor del profesor en el proceso de aprendizaje de los estudiantes. Estos sistemas poseen actividades de retroalimentación y autoevaluación que informan sobre los avances en la ejecución y los logros de los objetivos educacionales que persiguen los educandos (1). De igual modo el desarrollo de los Sistemas Tutoriales Inteligentes (STI) ha contribuido a la formación intelectual de los estudiantes en diversos centros educativos.

Los STI actúan de acuerdo a las necesidades personales de un estudiante. Son sistemas adaptables al conocimiento previo y a la capacidad de adquisición del conocimiento de una persona. Comenzaron a desarrollarse con la idea de transmitir el conocimiento de una forma inteligente, para poder asistir y guiar al estudiante en su proceso de aprendizaje. Identifican la manera en que el educando da solución a un problema, con el fin de poder brindarle ayudas cognitivas cuando lo requieran (2). Son sistemas que se adaptan al estado cognitivo de los estudiantes y que proporcionan la ayuda pedagógica adecuada para garantizar un buen aprendizaje (3). Son programas que portan conocimiento sobre cierta materia y cuyo propósito es transmitir este conocimiento a los estudiantes mediante un proceso interactivo individualizado, intentando simular la forma en que un tutor o profesor guiaría al alumno en el proceso de enseñanza-aprendizaje (4).

El término **inteligente** se refiere a la habilidad del sistema sobre qué enseñar, cuándo enseñar y cómo enseñar, simulando la actividad de un profesor real. Para lograrlo, un STI debe encontrar la información relevante sobre el proceso de aprendizaje de ese estudiante y aplicar el mejor medio de instrucción según sus necesidades individuales (2).

Cuba, inmersa en la tarea de alcanzar un mayor desarrollo económico y social, se ha trazado en los recientes Lineamientos de la Política Económica y Social del Partido

Comunista de Cuba, contar con mecanismos eficientes que permitan actualizar los programas de formación e investigación de las universidades para elevar la calidad y rigor del proceso docente-educativo (5).

En este sentido se han desarrollado STI en áreas como Teoría de Grafos, Análisis y Diseño de Sistemas y Estructura de Datos, desarrollados por la Facultad de Matemática, Física y Computación de la Universidad Central de las Villas (6). Además en la UCI se han implementado STI asignaturas como: Sistemas Operativos e Ingeniería de Software II (7) (8), los cuales sirven de apoyo al aprendizaje de los principales temas de estas asignaturas por parte de los estudiantes.

Situación problemática:

Otra de las herramientas desarrolladas en la UCI para mejorar el proceso de enseñanza-aprendizaje de sus estudiantes es el Entorno Virtual de Enseñanza-Aprendizaje (EVEA). El EVEA es un espacio de apoyo al proceso de formación de la Carrera de Ingeniería en Ciencias Informáticas, donde los profesores pueden implementar estrategias de enseñanza-aprendizaje complementarias a las clases presenciales, así como diseñar cursos semi-presenciales o totalmente a distancia. Sin embargo este sistema no permite realizar una orientación bibliográfica diferenciada de acuerdo a las necesidades particulares de un estudiante en la asignatura Redes y Seguridad Informática.

Por lo antes expuesto se plantea como **problema a resolver:**

¿Cómo recomendar la bibliografía adecuada a partir de las necesidades propias de los estudiantes, para complementar el proceso de auto-preparación de los mismos?

Para darle solución al problema anterior se propone como **objeto de estudio:** Los Sistemas Tutoriales Inteligentes, definiéndose como **campo de acción:** Los Sistemas Tutoriales Inteligentes para el proceso de enseñanza-aprendizaje de la asignatura Redes y seguridad informática en la Facultad 2 de la UCI.

Para ello se plantea como **objetivo general:** Desarrollar un modelo del profesor como parte de un Sistema Tutorial Inteligente que permita recomendar la bibliografía adecuada a partir del nivel de conocimiento de los estudiantes para la auto-preparación de los mismos.

Como **objetivos específicos** para alcanzar la meta propuesta, se encuentran:

- Realizar un estudio teórico conceptual sobre los STI para personalizar el proceso de enseñanza-aprendizaje de los estudiantes.
- Diseñar el Modelo del profesor como parte de un Sistema Tutorial Inteligente para una mejor comprensión de su funcionamiento.
- Implementar el Modelo del profesor como parte de un Sistema Tutorial Inteligente con el fin de recomendar la bibliografía adecuada para la auto-preparación de los estudiantes.
- Validar la propuesta de solución para comprobar su correcto funcionamiento.

Para dar cumplimiento a los objetivos propuestos, se definen las siguientes **tareas de la investigación**:

- Descripción de las características de los STI para una mejor comprensión de su funcionamiento.
- Comparación de diferentes tipos de Sistemas Basados en el Conocimiento (SBC) para fundamentar el paradigma a utilizar en el desarrollo de la propuesta de solución.
- Descripción de las metodologías, lenguaje y herramientas a utilizar en el desarrollo del STI para fundamentar su selección.
- Diseño de las historias de usuario para orientar la implementación del Modelo del profesor como parte de un Sistema Tutorial Inteligente.
- Implementación de las funcionalidades del Modelo del profesor como parte de un Sistema Tutorial Inteligente para recomendar de forma diferenciada la bibliografía adecuada a los estudiantes en la asignatura Redes y Seguridad Informática.
- Realización de las pruebas de funcionamiento del Modelo del profesor como parte de un Sistema Tutorial Inteligente para verificar su correcto funcionamiento.

Métodos Teóricos:

Método Histórico-Lógico: Permite una mejor comprensión de la investigación mediante el estudio y profundización de los STI en el mundo, así como sus principales características e impacto en el proceso de enseñanza-aprendizaje.

Método de Modelación: Este método tiene la capacidad de representar las características y las relaciones fundamentales del sistema, permitiendo desarrollar los diagramas necesarios durante su desarrollo.

Métodos Empírico:

La entrevista: Este método permite realizar entrevistas a profesores que imparten la asignatura Redes y Seguridad informática de la Facultad 2 de la UCI, con el objetivo de obtener información necesaria con respecto a los requerimientos que el sistema debe tener.

La tesis consta de una introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos que ampliarán la información que se brinda en la investigación.

Capítulo 1: Fundamentos Teóricos que Sustentan el Desarrollo de los Sistemas Tutoriales Inteligentes.

En este capítulo se hace un estudio sobre cómo contribuye el proceso de enseñanza-aprendizaje a la formación intelectual de los hombres. De igual manera se aborda sobre los STI y sus componentes fundamentales. Por otra parte se hace un análisis sobre los tipos de SBC existentes en la actualidad, así como sus principales características para fundamentar el paradigma a utilizar en el desarrollo de la propuesta de solución. Se describen las herramientas, lenguajes de programación y metodología de desarrollo que serán empleadas en el desarrollo del sistema.

Capítulo 2: Características del STI

En este capítulo se refleja el impacto social desde el punto de vista del campo Ciencia, Tecnología y Sociedad (CTS). Se describe el funcionamiento del sistema propuesto, definiendo las listas de reserva del producto, así como las fases de Exploración y Planificación siguiendo la metodología XP, confeccionando las Historias de Usuarios (HU) de cada iteración y los tiempos de entrega de cada una.

Capítulo 3: Diseño, Implementación y Prueba.

En este capítulo se define la arquitectura del sistema, los patrones de diseño utilizados, así como las Tarjetas Clase-Responsabilidad-Colaborador y las Tareas de ingeniería generadas por cada HU. Además se muestra el Diagrama de despliegue y

de Modelo de datos de la aplicación. De igual modo se reflejan los resultados de las pruebas realizadas a la propuesta de solución, garantizando el cumplimiento del objetivo trazado en el desarrollo del sistema.

CAPÍTULO I. Fundamentos Teóricos que Sustentan el Desarrollo de los Sistemas Tutoriales Inteligentes.

1.1 Introducción

En este capítulo se hace referencia a los STI, sus componentes fundamentales, arquitectura y los referentes actuales en el mundo, Cuba y la UCI. Por otra parte se hace un análisis sobre los tipos de SBC existentes en la actualidad, así como sus principales características para fundamentar el paradigma a utilizar en el desarrollo de la propuesta de solución. Se describen las herramientas, lenguajes de programación y metodología de desarrollo que serán empleadas en el desarrollo del sistema.

1.2 Sistemas Tutoriales Inteligentes

Un STI es un sistema de enseñanza asistida por computadora que utiliza técnicas de Inteligencia Artificial principalmente para representar el conocimiento y dirigir una estrategia de enseñanza.

Es capaz de comportarse como un experto, tanto en el dominio de conocimiento que enseña, como en el dominio pedagógico, donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje.

Considerando que los estudiantes poseen diferentes estilos o formas de aprendizaje, es importante que un STI sea capaz de ofrecer instrucción individualizada, es decir, debe ser capaz de adaptar sus instrucciones para satisfacer los estilos individuales de cada aprendiz. Este es el principio general que rige la concepción de un STI (9).

1.2.1 Componentes fundamentales de un Sistema Tutorial Inteligente:

Los STI están compuestos por un módulo del dominio, un módulo del estudiante y el módulo pedagógico, que operan de forma interactiva y se comunican a través de un módulo central que se denomina módulo entorno (2).

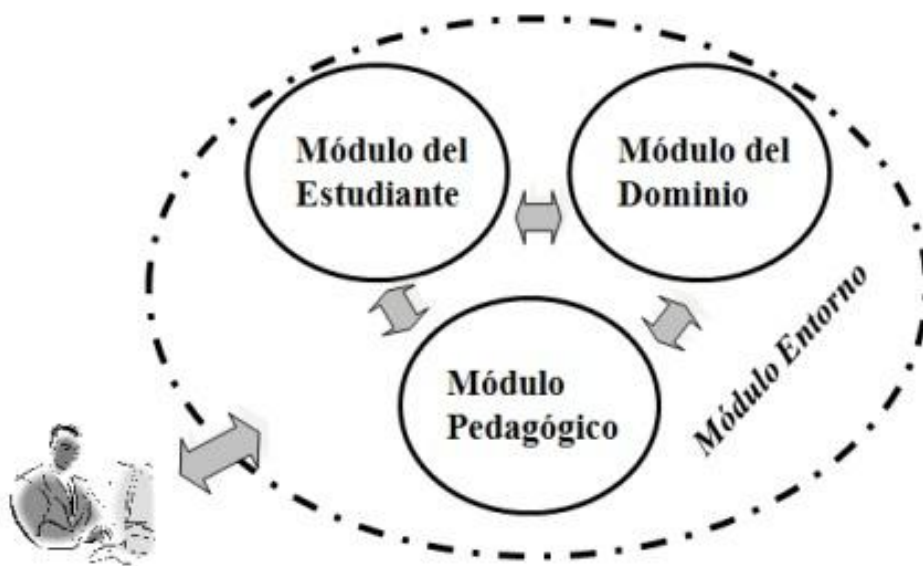


Figura 1: Arquitectura de un STI.

El Módulo del Estudiante

El módulo del estudiante es de gran importancia para el desarrollo de un STI, producto a que el mismo se realiza con el fin de obtener una representación completa y precisa de las características del estudiante, para garantizar una correcta orientación del mismo durante el proceso de enseñanza-aprendizaje (2).

Entre las principales características del Módulo se encuentran:

- **Estilos de aprendizaje**

Conjunto de características psicológicas, rasgos cognitivos, afectivos y fisiológicos que suelen expresarse conjuntamente cuando una persona debe enfrentar una situación de aprendizaje. Los rasgos cognitivos tienen que ver con la forma en que los estudiantes estructuran los contenidos, forman y utilizan conceptos, interpretan la información y resuelven los problemas. Los rasgos afectivos se vinculan con las motivaciones y expectativas que influyen en el aprendizaje, mientras que los rasgos fisiológicos están relacionados con el biotipo y el biorritmo del estudiante (2).

- **Nivel de conocimiento**

Características propias de cada estudiante referente al grado de conocimiento que posee acerca de conceptos, temas y asignaturas (2).

- **Información personal**

Datos como la edad, género, idioma y otras informaciones que puedan ser de interés (2).

El Módulo del Dominio

Satisface dos propósitos diferentes, en primer lugar, presentar la materia de la forma adecuada para que el estudiante adquiera las habilidades y conceptos, lo que incluye la capacidad de generar preguntas, explicaciones, respuestas y tareas para el estudiante. En segundo lugar, el módulo del dominio debe ser capaz de resolver los problemas generados, corregir las soluciones presentadas y aceptar aquellas soluciones válidas que han sido obtenidas por medios distintos. En este módulo, el conocimiento debe organizarse pedagógicamente para facilitar el proceso de enseñanza-aprendizaje (2).

El Módulo Entorno

Permite la interacción del estudiante con el sistema. Los entornos deben ser fáciles de utilizar y atractivos, de forma que el estudiante pierda el mínimo tiempo posible en aprender a utilizar el sistema y pueda centrar toda su atención en el contenido.

La necesidad de crear ambientes computacionales capaces de mantener el interés de sus usuarios, implica el desarrollo de interfaces personalizadas para ofrecer un servicio individualizado al usuario, adaptando su interacción con el sistema a sus necesidades e intereses personales (2).

El Módulo Pedagógico

Decide qué, cómo y cuándo enseñar los contenidos, adaptando sus decisiones pedagógicas a las necesidades del estudiante, es el encargado de comparar las características de los estudiantes con el contenido a enseñar y elegir la mejor forma de tomar las decisiones pedagógicas oportunas (2).

En la presente investigación el desarrollo de este módulo permite recomendar la bibliografía adecuada a un estudiante a partir del nivel de conocimiento evidenciado por el mismo luego de resolver un cuestionario existente en el sistema.

1.3 Referentes sobre STI en el Mundo, Cuba y en la Universidad de las Ciencias Informáticas

Entre los STI desarrollados en el mundo se encuentra, el sistema **AGT** (*Advanced Geometry Tutor*), el mismo se utiliza para el aprendizaje de temas sobre la geometría avanzada, desarrollado en la *University of Pittsburgh* a través de la *National Science Foundation* y el *Center for Interdisciplinary Research on Constructive Learning Environments* de la *University of Pittsburgh* y *Carnegie Mellon University* (10). Se basa en comparar dos estrategias de resolución de problemas, encadenamiento hacia adelante y encadenamiento hacia atrás, determinando el nivel de adquisición del conocimiento de los estudiantes, con el fin de definir cuál estrategia acelera más el aprendizaje.

En el año 2012 se desarrolló en la Universidad de Concepción, Chile el sistema **ELE-TUTOR**, el mismo es un Sistema Tutorial Inteligente para el Español como Lengua Extranjera. Cuenta con un analizador automático que garantiza identificar y alertar a un estudiante sobre un error cometido durante la redacción de una respuesta a interrogantes que muestre el sistema. Para ello **ELE-TUTOR** tiene la capacidad de predecir los errores que puede cometer un usuario en un momento determinado de su aprendizaje. Lo que conllevó a almacenar los errores más frecuentes cometidos por los estudiantes con lengua materna inglesa en clases presenciales de español (11).

De igual forma en Cuba se han desarrollado STI que permiten personalizar el proceso de enseñanza-aprendizaje. Un ejemplo de ello es el Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos (12) y el Sistema para el Montaje de un Tutorial Inteligente para personalizar las tareas docentes teniendo en cuenta los conocimientos de los estudiantes en la asignatura Ingeniería de Software II (13), desarrollados en la UCI.

El desarrollo de estos sistemas les ha permitido a los estudiantes contar con medios de enseñanza lo suficientemente adaptables a su conocimiento y capacidad de evolución de aprendizaje en los temas tratados por estas aplicaciones. Sin embargo,

ninguno de estos es adaptable para dar solución al problema planteado en la presente investigación.

El sistema AGT solo facilita crear habilidades en los estudiantes para la resolución de problemas de Geometría Avanzada. El ELE-TUTOR se desarrolló con el objetivo de complementar el aprendizaje del español como idioma extranjero, ya que este sistema contiene un analizador automático que permite detectar los errores de redacción cometidos por los estudiantes. De igual modo el Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos cuentan con un solo cuestionario que incluye todos los temas de la asignatura, sin tener en cuenta que los estudiantes pueden enfrentarse a preguntas relacionadas con temas de las asignaturas que no han sido impartidos en clases presenciales y en la presente investigación no se busca sustituir la labor del profesor sino complementarla, este sistema además no está disponible para evaluar la posibilidad de adaptarlo y lograr el cumplimiento del objetivo de la presente investigación. Por otro lado el Sistema para el Montaje de un Tutorial Inteligente para personalizar las tareas docentes teniendo en cuenta los conocimientos de los estudiantes en la asignatura Ingeniería de Software II, presenta una gran dependencia con el profesor de la asignatura, ya que para que un estudiante obtenga los resultados alcanzados luego de realizar un cuestionario tiene que esperar a que el profesor lo califique e inserte la evaluación en el sistema. No obstante el estudio realizado de los mismos sirvió de base para el desarrollo de la propuesta de solución, promoviendo la incorporación de aspectos comunes que presentan estas herramientas que resultan de utilidad, como es el caso de las características que presenta el Módulo Pedagógico en cada uno de ellos.

1.4 Sistemas basados en el conocimiento

Los Sistemas Basados en el Conocimiento (SBC), son sistemas computacionales capaces de soportar la representación explícita del conocimiento de un dominio específico y de explotarlo a través de los mecanismos apropiados de razonamiento, para proporcionar un comportamiento de alto nivel en la resolución de problemas. No todos los paradigmas para crear SBC, facilitan la concepción de un STI, pues en el desarrollo de un STI es determinante la forma de representación del conocimiento en

cada uno de sus módulos y a partir de dicho conocimiento diagnosticar al estudiante para que el sistema se adapte a sus características (6).

Sin embargo, existen similitudes entre los STI y los SBC, una de ellas es la separación del conocimiento (Base del Conocimiento) del método de solución del problema o Máquina de Inferencia (MI) (6).

En la base del conocimiento se almacena el conocimiento necesario para resolver los problemas del dominio. El conocimiento puede ser de diferentes tipos:

- Conocimiento simbólico sobre cómo resolver los problemas, el cual se puede representar mediante diversas formas (reglas de producción, redes semánticas).
- Probabilidades o frecuencias que modelan cómo se relacionan los valores de los diferentes rasgos que caracterizan el dominio.
- Pesos de una red neuronal
- Casos o ejemplos de problemas del dominio.

Los diferentes tipos de conocimiento dan lugar a diferentes tipos de sistemas basados en el conocimiento, entre ellos:

- Sistemas basados en reglas.
- Sistemas basados en probabilidades.
- Sistemas expertos conexionistas o redes expertas.
- Sistemas basados en casos (6).

La MI, tiene implementado un método que utiliza el conocimiento de la base del conocimiento para resolver los problemas del dominio. El tipo de conocimiento determina qué método de solución de problemas será posible utilizar (6).

1.4.1 Ingeniería del conocimiento en el desarrollo de sistemas basados en el conocimiento

La Ingeniería del Conocimiento (IC) surge como consecuencia de la necesidad de establecer principios metodológicos y científicos que permitan desarrollar sistemas basados en el conocimiento a partir de los fundamentos de la informática en general y de la inteligencia computacional en particular. En este aspecto puede verse como

la especialización de Ingeniería de Software en su aplicación al desarrollo de Sistemas Inteligentes (2).

La IC se enfoca al desarrollo de sistemas basados en el conocimiento, destacándose la necesidad de la adquisición del conocimiento así como su especificación, verificación, validación, diseño e implementación en sistemas informáticos o lenguajes apropiados para la construcción de bases de conocimiento para la toma de decisiones (2).

1.4.2 Sistemas basados en reglas

Los Sistemas Basados en Reglas, son sistemas basados en el conocimiento, en el que se hace una representación mediante reglas de producción o reglas condicionales. Las reglas representan el conocimiento utilizando el formato:

SI-ENTONCES (IF-THEN)

SI (IF), es el antecedente, premisa, condición o situación.

ENTONCES (THEN), es el consecuente, conclusión, acción o respuesta. (6)

En este tipo de sistema, la base de hechos representa el conocimiento en las variables de entrada y salida del sistema y en la base de reglas se combinan variables de la base de hechos con el IF y el THEN junto con los operadores lógicos AND y OR. La parte de una regla que se encuentra entre IF y THEN es el antecedente y el resto el consecuente. En el antecedente es donde están las premisas que se deben cumplir para que la regla se pueda aplicar y en el consecuente el conjunto de acciones derivadas de la aplicación de la regla.

La MI utiliza los datos y el conocimiento para obtener conclusiones. Por ejemplo, si la premisa de una regla es cierta, entonces la conclusión de la regla también debe ser cierta. Para obtener estas conclusiones se utilizan diferentes tipos de estrategias de inferencia y control. Entre las reglas de inferencia se encuentran Modus Ponens y Modus Tollens y en cuanto a las estrategias de inferencia se encuentra, el encadenamiento de reglas hacia delante o dirigida por datos y hacia atrás o dirigida por objetivo, cuyas reglas son utilizadas por la MI para obtener conclusiones simples,

que son las que resultan de una regla simple y las compuestas, que son las que resultan de más de una regla (6).

1.4.3 Sistemas basados en probabilidades

La base de conocimiento se forma por el espacio probabilístico que describe el problema. En este tipo de sistemas la MI está basada en probabilidades condicionales y se encarga de actualizar las probabilidades con base en los hechos que se observen del ambiente en que se desempeña y cuenta con el sistema de control de coherencia que logra la coherencia en el sistema, especificando para ello las restricciones que deben cumplir los datos introducidos. El método de solución que se emplea es el teorema de Bayes para calcular probabilidades condicionales (6).

1.4.4 Sistemas expertos conexionistas o redes neuronales artificiales

Las redes neuronales artificiales (RNA) son modelos matemáticos inspirados en las neuronas biológicas del cerebro humano. Con estos modelos se resuelven múltiples problemas de reconocimiento, aproximación, predicción, clasificación y optimización (14).

Un Sistema Experto (SE) es un conjunto de programas que, sobre una BC, posee información de uno o más expertos en un área específica. Un SE que utiliza redes neuronales (RN) es mejor conocido como sistema experto conexionista (SEC). El conexionismo se establece como un paradigma que permite modelar el funcionamiento del sistema nervioso a partir de formas simplificadas del funcionamiento de las neuronas que lo constituyen y de las conexiones que entre ellas se establecen (15).

1.4.5 Sistemas basados en casos

Las raíces de los Sistemas Basados en Casos hay que buscarlas en ciertos resultados de psicología donde se muestra que, en muchas ocasiones, los seres humanos resuelven problemas sobre la base de experiencias pasadas y no a partir de un conocimiento profundo del problema en cuestión. Los médicos, por ejemplo, buscan conjuntos de síntomas conocidos, los ingenieros toman muchas de sus ideas de soluciones previas ya construidas con éxito para dar solución a nuevas interrogantes.

El método de solución de problemas que utilizan estos sistemas es el Razonamiento Basado en Casos (RBC). Al enfrentarse a nuevas situaciones lo primero que se hace es buscar en la memoria, experiencias ya vividas que sean similares al nuevo problema y a partir de esto, se establecen semejanzas y diferencias que se ajustan a las soluciones dadas anteriormente para obtener una nueva solución (16) .

Cuando existe la necesidad de resolver un nuevo problema, éste es descrito para el módulo de recuperación, el cual realiza una búsqueda en la BC y encuentra problemas o casos similares. Los problemas o casos similares resueltos son recuperados y enviados al módulo de adaptación, donde son analizados para construir una solución para el nuevo problema y al encontrarse la solución se almacena junto con la descripción del problema en la BC, ya que constituye un nuevo caso (16).

El enfoque que utilizan los Sistemas Basados en Casos para la adquisición de conocimiento es una de las ventajas que se le acreditan a este tipo de sistemas; pues razonan desde episodios específicos, lo cual evita el problema de descomponer el conocimiento del dominio y generalizarlo en reglas. Otras de las ventajas de los SBC están fundamentadas en la flexibilidad para representar el conocimiento a través de los casos, la organización de la BC y de las estrategias de recuperación y adaptación de los casos. Ventajas lo son también, el reuso de las soluciones previas para resolver un nuevo problema (2).

Módulos de un Sistema Basado en Casos

Los módulos fundamentales de un Sistemas Basados en Casos para el desarrollo de un STI son:

- Módulo de recuperación.
- Módulo de adaptación.
- Módulo de almacenamiento (2) .

Módulo de recuperación

El módulo de recuperación constituye la primera etapa en el ciclo de funcionamiento de un SBC. Este módulo exige realizar una comparación entre el nuevo caso y los casos de la base de caso. Este proceso está determinado por las funciones de

semejanzas que se utilizan al comparar los rasgos predictores, las cuales se seleccionan de acuerdo a sus valores. Para evitar la recuperación de casos que resulten irrelevantes para la solución, se utiliza un umbral de semejanza que reduzca el número de casos recuperados a solo los más parecidos (17).

La función de semejanza permite determinar cuán semejante es el nuevo caso respecto a los existentes en la BC. No existe una función única para todos los dominios, por lo que la eficiencia del proceso de recuperación de los casos semejantes radica en su correcta selección. La misma permite la recuperación de aquellos casos que presenten mayor semejanza con el nuevo. Puede incluir la importancia o relevancia de cada rasgo al integrar los valores obtenidos en la comparación rasgo a rasgo, de modo que el resultado de la comparación de rasgos más relevantes, tenga más peso en el cálculo del grado de similitud entre el caso nuevo y un caso de la BC (17).

Módulo de adaptación

Después de seleccionados y recuperados los casos más semejantes, las soluciones que proponen dichos casos pueden utilizarse directamente como solución al nuevo problema o si fuese necesario se realiza una modificación de éstas, este método de solución del problema es conocido como adaptación de las soluciones.

Un adaptador de casos toma la descripción de un problema actual, la descripción de un problema similar y la solución a este, para construir con estos elementos la solución del problema actual (17).

El proceso de adaptación se compone de dos fases: identificación de aquello que hay que adaptar y aplicación de un método de adaptación. Además en éste se pueden seguir dos estrategias: adaptación estructural, que consiste en aplicar las reglas de adaptación directamente a la solución almacenada en un caso, y adaptación derivacional que consiste en utilizar y a la vez modificar las reglas que generaron la solución original, partiendo de las características del nuevo problema para generar la solución a éste (17).

Los principales métodos de adaptación son:

Adaptación nula: Consiste en transferir la solución que propone un problema similar recuperado directamente como solución al problema actual (17).

Adaptación transformacional: Consiste en reutilizar soluciones recuperadas después de aplicárseles determinadas transformaciones que generen la solución buscada. Estas transformaciones se pueden realizar ajustando parámetros o substituyendo valores (17).

Adaptación derivacional: Consiste en realizar la adaptación de algún elemento de la solución que no se ajusta correctamente, reaplicando para el problema actual, el procedimiento que fue utilizado para calcular dicho elemento de la solución en el caso previo (17).

Sistemas cooperativos: Al adaptar utilizando esta técnica se establece una cooperación entre el sistema y sus usuarios, donde el sistema únicamente presenta las soluciones relevantes y el experto humano es quien se encarga de realizar la adaptación de éstas (17).

Módulo de almacenamiento

En este Módulo se almacena el caso recién resuelto en la BC para ser utilizado en la resolución de futuros problemas (2).

Tipos de sistemas basados en casos

Sistemas interpretativos: La interpretación se usa cuando el problema no está bien comprendido y cuando hay necesidad de criticar una solución. Estos sistemas toman una situación o solución como entrada y su salida es un argumento fundamentando la solución. Son útiles para sistemas de clasificación, evaluación de una solución, argumentación, justificación de una solución, interpretación y predicción de los efectos de una decisión o plan (17).

Sistemas resolvedores de problemas:

El estilo de solución de problemas se caracteriza por la gran capacidad que presenta el sistema para realizar el proceso de adaptación mediante el cual se construye la solución del nuevo problema a partir de la solución dada a un problema semejante (17).

Fundamentos de la selección

Para la selección del SBC a utilizar en la investigación, se tuvo en cuenta la envergadura del proceso de adquisición del conocimiento de los diferentes tipos de SBC. En los Sistemas Basados en Reglas este proceso es complejo y prolongado, pues la extracción del conocimiento se refiere a la formalización de reglas y en la presente investigación sería necesario la construcción de una gran cantidad de reglas para representar el conocimiento adecuado para evaluar un estudiante y brindarle una bibliografía adecuada de acuerdo a sus necesidades.

En los Sistemas Basados en Probabilidades, la adquisición del conocimiento consiste en coleccionar muestras y realizar un procesamiento estadístico que produzca las probabilidades o frecuencias que forman la base de conocimiento. No son factibles para todo tipo de dominio, pues se dificulta calcular las probabilidades con ayuda de expertos humanos cuando existen carencias de conocimiento.

En las Redes Neuronales Artificiales la adquisición del conocimiento incluye la selección de los ejemplos, el diseño de su topología y el entrenamiento de la red para hallar el conjunto de pesos. Facilitan el trabajo con información incompleta y brindan algoritmos poderosos de aprendizaje para crear la base de conocimiento; pero requieren de muchos ejemplos y son cajas negras que no explican cómo la solución se alcanza y en la presente investigación resulta de utilidad que un profesor conozca las causas que provocaron la recomendación de una bibliografía a un estudiante.

Sin embargo en los Sistemas Basados en Casos la adquisición del conocimiento se reduce a la selección de un conjunto de ejemplos o casos resueltos y su organización en la base de casos. Estos sistemas razonan desde episodios específicos, lo cual evita el problema de descomponer el conocimiento del dominio. Además en la presente investigación la base de conocimiento puede ser estructurada a partir de la organización de respuestas brindadas por estudiantes a pruebas realizadas en la asignatura Redes y Seguridad Informática. Todo esto motivó al equipo de desarrollo a utilizar los Sistema Basado en Casos como técnica de la Inteligencia Artificial para el desarrollo de la investigación.

1.5 Metodología del desarrollo

Todo proceso de desarrollo de software necesita de un hilo conductor que guíe al equipo de desarrollo durante todo el ciclo de vida del proyecto, para garantizar así la obtención de resultados satisfactorios. Con este fin y ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental en el momento de desarrollar un producto de software, surgen las metodologías de desarrollo.

Las metodologías de desarrollo se clasifican en:

Las metodologías tradicionales

Estas metodologías generan mucha documentación, están destinadas para un equipo de desarrollo grande y para proyectos a largo plazo. Además, no están definidas para cambios frecuentes y tratan de especificar todos los requisitos del software antes de comenzar su desarrollo, lo cual conlleva a la decisión de utilizar las metodologías ágiles en la presente investigación (18).

Las metodologías ágiles

Estas metodologías están orientadas a la interacción con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Son efectivas en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad (18).

Teniendo en cuenta la clasificación de las metodologías expuestas con anterioridad, las características del sistema que se desea desarrollar y el corto tiempo para su implementación, se decide utilizar como guía para el proceso de desarrollo una metodología ágil.

Actualmente existen varias metodologías ágiles capaces de adaptarse a las características del sistema que se desea desarrollar, entre estas se destacan eXtreme Programming (XP), SCRUM Y Proceso Unificado Ágil (OpenUP).

El equipo de desarrollo decide utilizar XP, pues es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado, genera poca documentación por ser un proyecto

pequeño. Además si durante el desarrollo del sistema, se detecta la necesidad de cambiar funcionalidades, se realiza un acuerdo con el cliente, se ajusta el plan de iteraciones y se toma una nueva dirección en el desarrollo, lo cual resulta conveniente pues si se realizan cambios al sistema luego de una posible entrega no sería necesario arreglar tanta documentación (19).

1.6 Lenguaje de programación del lado del servidor

El equipo de desarrollo decide utilizar PHP en su versión 5.3 como lenguaje de programación del lado del servidor ya que el mismo es un lenguaje adecuado para el desarrollo web y embebido en páginas HTML. El principal objetivo del lenguaje es permitir que los desarrolladores web escriban páginas generadas dinámicamente de forma rápida. Tiene soporte para muchos tipos de gestores de bases de datos, entre los principales están MySQL, PostGreSQL y Oracle. El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador, lo que garantiza que la programación en PHP sea segura y confiable (20).

1.7 Lenguaje de Programación del lado del Cliente

JavaScript

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas, incorporando efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (21).

Librerías de JavaScript

Las aplicaciones cada vez tienen más código JavaScript y son utilizadas más librerías para gestionar mejor el aspecto visual de la aplicación. Por eso, librerías como jQuery, Prototype o Dojo han entrado en auge. Normalmente estas librerías tienen una versión de desarrollador y una versión comprimida para ponerla en producción y reducir la carga y transferencia del código JavaScript (21).

En la presente investigación, se decide utilizar JQuery, ya que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones. Es la librería JavaScript más utilizada actualmente, gratuita, de código abierto. Entre las ventajas de utilizar esta librería se encuentran reducción y reutilización de código que permite realizar scripts con dinamismos sencillos, fácil manejo de animaciones, permitiendo desarrollar componentes dinámicos con los que se enriquecen las aplicaciones.

CSS3

CSS3 es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML, constituye la mejor forma de separar los contenidos y su presentación, además es imprescindible para crear páginas web complejas. Esta versión presenta como principales características, mayor control de los estilos de los elementos del sistema y mayor número de efectos visuales (22).

HTML5

Es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento. Un documento hipertexto no sólo se compone de texto, puede contener imágenes, sonido o videos. Los documentos HTML son visualizados en los navegadores, los mismos son los encargados de interpretar el código HTML de los documentos, y de mostrar a los usuarios las páginas web resultantes del código interpretado (23).

1.8 Framework web

Un Framework o Marco de Trabajo para Aplicaciones Web es un tipo de framework diseñado para soportar el desarrollo de sitios dinámicos, aplicaciones, servicios y recursos para la web. Este tipo de framework provee por lo general toda la funcionalidad central a la mayoría de las aplicaciones web, tales como administración de la sesión del usuario, persistencia de datos y sistemas de plantillas.

Representa un conjunto de componentes que forman un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. Los objetivos principales que persigue un

Framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo de software (24).

Se decide utilizar **Symfony2** como framework web ya que el mismo ha sido ideado para explotar al límite todas las características de PHP 5.3. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajan en el sistema a desarrollar. Además está diseñado para optimizar de acuerdo a sus características, el desarrollo de las aplicaciones Web. Emplea el tradicional patrón arquitectónico Modelo-Vista-Controlador (MVC) para separar las distintas partes que forman una aplicación web. Fácil de instalar y de configurar (25).

De igual modo se utiliza **Bootstrap** ya que el mismo es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ofrece una serie de plantillas CSS y ficheros JavaScript que permiten integrar el framework de forma sencilla y potente en las aplicaciones web, permite crear interfaces que se adapten a los diferentes navegadores y además se integra fácilmente con las principales librerías JavaScript como JQuery (26).

1.9 Herramienta de modelado

EL equipo desarrollo decide utilizar **Visual Paradigm 8.0**, pues permite modelar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Soporta el lenguaje de modelado UML. Fácil de instalar y actualizar. Está disponible en versiones de Windows y Linux. Constituye una herramienta de software libre de probada utilidad para los Ingenieros de Software, Analistas de Sistemas y Arquitectos (27).

1.10 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar la definición de los datos, su mantenimiento, su control, su privacidad y su manipulación. Entre los SGBD más utilizados se encuentran **MySQL** y **PostgreSQL**.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. (28)

Características de MySQL

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

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas.
- Gestión de usuarios y passwords, manteniendo un buen nivel de seguridad en los datos (28).

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley.

Características de PostgreSQL

A continuación se enumeran las principales características de este gestor de bases de datos:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos (29).

El equipo de desarrollo decide utilizar el SGBD **PostgreSQL** en su versión 9.4.1 ya que a pesar de que MySQL es más rápido que PostgreSQL a la hora de resolver consultas, y de tener mejor documentación y herramientas de administración, PostgreSQL ofrece una garantía de integridad mucho más fuerte que MySQL.

Además, presenta una mejor escalabilidad y rendimiento bajo grandes cargas de trabajo. (30)

1.11 Servidor Web

Un servidor Web consta de un intérprete HTTP el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla.

El equipo de desarrollo decide utilizar el servidor web Apache en su versión 2.2.4 pues el mismo:

- Permite la creación de sitios web dinámicos utilizando PHP.
- Es flexible, rápido y eficiente.
- Continuamente actualizado y adaptado a los nuevos protocolos.
- Es Multiplataforma.
- Es una tecnología gratuita de código fuente abierto.
- Es un servidor altamente configurable (31).

1.12 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE) es una aplicación compuesta por un conjunto de herramientas útiles para un desarrollador de software. Puede ser exclusivo para un lenguaje de programación o bien, poder utilizarse para varios. Está compuesto por un editor de código, un compilador y herramientas de automatización de la compilación, un depurador y en algunos casos un constructor de interfaz gráfica (32).

El equipo de desarrollo decide utilizar como Entorno de Desarrollo Integrado (IDE): El **NetBeans** en su versión 8.0, ya que el mismo está desarrollado para usarse generalmente con lenguaje Java, aunque permite su uso con otros lenguajes de programación como PHP. NetBeans es cómodo para los programadores. Tiene un excelente balance entre una interfaz con múltiples opciones y un aceptable completamiento de código, facilitando el trabajo para la creación de aplicaciones web con rapidez. Es un IDE multiplataforma, de código abierto y fácil de instalar (33).

1.13 Conclusiones parciales

Durante la realización de este capítulo se realizó un estudio sobre las principales características de los STI, así como de los SBC, adoptando el paradigma de creación de estos para el desarrollo de la solución propuesta. Se realizó la selección de la metodología, las tecnologías y herramientas necesarias para el desarrollo del sistema.

CAPÍTULO 2. Características del STI.

2.1 Introducción

En este capítulo se hace referencia al impacto social del Modelo del profesor como parte de un Sistema Tutorial Inteligente para recomendar de forma diferenciada la bibliografía adecuada a los estudiantes en la asignatura Redes y Seguridad Informática, de igual modo se describe la propuesta de solución del sistema, se definen las listas de reserva del producto, se detallan las fases de Exploración y Planificación de la metodología XP, las cuales permiten realizar la confección de las Historias de Usuarios (HU) para cada iteración y además la definición de los tiempos de entrega de cada una de las iteraciones.

2.2 Impacto social del Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura Redes y Seguridad Informática.

La ciencia, la tecnología y la sociedad son procesos que no se pueden desarrollar por separado, pues uno complementa al otro y esto determina la interrelación que poseen como conceptos aplicados al desarrollo del software. A lo largo de la historia, el desarrollo de la ciencia y la tecnología ha transformado las formas de vida de las sociedades, del mismo modo que las formas de vida social han sido también determinantes para el desarrollo de la Ciencia y de la Tecnología.

El desarrollo del Modelo del profesor como parte de un Sistema Tutorial Inteligente para recomendar de forma diferenciada la bibliografía adecuada a los estudiantes en la asignatura Redes y Seguridad Informática, se enfoca en las perspectivas que siguen los Lineamientos de la Política Económica y Social del Partido Comunista de Cuba, de desarrollar productos que respondan a las necesidades del país de actualizar los programas de formación e investigación de las universidades, para elevar la calidad y rigor del proceso docente-educativo en función de alcanzar un mayor desarrollo económico y social (5).

Los efectos del impacto están encaminados a:

- Desarrollar una herramienta informática que permita evaluar el dominio de un determinado objetivo de la asignatura Redes y Seguridad Informática por parte de los estudiantes.

- Garantizar una orientación bibliográfica diferenciada de acuerdo a las necesidades de los estudiantes en la asignatura Redes y Seguridad Informática, como apoyo al proceso de enseñanza-aprendizaje de forma autónoma.
- Fortalecer la auto-preparación de los estudiantes en la asignatura con vista a las evaluaciones frecuentes, parciales y finales.

2.3 Propuesta del Sistema

Para el desarrollo del Modelo del profesor como parte de un Sistema Tutorial Inteligente para recomendar de forma diferenciada la bibliografía adecuada a los estudiantes en la asignatura Redes y Seguridad Informática, se implementa el Módulo Pedagógico de un STI, el cual cuenta con un conjunto de preguntas por cada tema de la asignatura Redes y Seguridad Informática, las que permiten obtener el nivel de conocimiento que presenta un estudiante al responder las mismas y así verificar el cumplimiento de un determinado objetivo de la asignatura y recomendar la bibliografía más idónea para el proceso de auto-preparación de un estudiante en caso de ser necesario.

De igual manera se utiliza el RBC, en el cual, los casos en la BC reflejan un conjunto de posibles respuestas a las preguntas relacionadas con cada tema de la asignatura, además de reflejar el estado cognitivo de un estudiante, los casos contienen la bibliografía a estudiar por dicho estudiante. El sistema contará con un cuestionario por cada tema, luego de que el estudiante responde las preguntas asociadas al cuestionario, el sistema realiza el RBC con los valores que representan las respuestas a las preguntas (nuevo caso) y los casos en la BC para recomendar a un estudiante la bibliografía asociada al caso más semejante en la BC

La Base de Casos

La BC permite la recuperación de los casos semejantes al nuevo caso analizado, contiene las experiencias, ejemplos o casos a partir de los cuales el sistema hace sus inferencias. Esta base puede ser generada a partir de casos o ejemplos resultantes del trabajo de expertos humanos (2) .

Existen gran cantidad de variantes en la forma que se organizan las BC. Estas variantes se pueden agrupar en dos categorías principales: la plana y la jerárquica.

Memoria plana:

En esta estructura se presentan los casos completos de forma secuencial utilizando algún tipo de indexación ya sea manual o automática. Tiene como desventaja que la búsqueda en la misma es costosa computacionalmente, por lo que es recomendable para BC pequeñas. Por otra parte la inserción de nuevos casos en este tipo de estructuras es muy sencilla ya que basta con incluir un nuevo registro con el nuevo caso respetando el método de indexación que se esté utilizando (17).

Memoria jerárquica:

En una estructura con memoria jerárquica se utilizan representaciones en forma de árbol, en los que cada nodo interior representa un atributo del caso y en las hojas se almacenan las soluciones a los mismos. Esta estructura sacrifica sencillez en la inserción de los nuevos casos por eficiencia en los algoritmos de búsqueda (17).

Para seleccionar la estructura de la BC, se tuvo en cuenta la importancia que tiene la información que presenta cada modelado del estudiante, por lo que la estimación de un determinado material didáctico debe comprender el análisis de todos los casos necesarios, lo que no requiere de estructuras que modelen entradas de datos incompletos. Producto a esto se seleccionó una estructura plana para la BC.

Los casos en la BC representan el estado del conocimiento y comportamiento del estudiante, así como el entrenador o material didáctico más adecuado. Cada caso es un ejemplo de modelado de estudiante, el cual se divide en modelo del estudiante (rasgos predictores) y materiales didácticos más adecuados para ese modelo (rasgo objetivo) (2).

2.3.1 Fase de recuperación

En la fase de recuperación se seleccionan de la BC los casos más semejantes al nuevo caso. Para alcanzar una mayor exactitud a la hora de seleccionar los casos, se define un umbral que representa la cota inferior de los valores de semejanzas obtenidos entre el nuevo caso y los existentes en la BC, el valor del umbral definido

es 0.75 para lograr que el caso recuperado de la BC sea lo más semejante posible al problema a resolver.

Función de semejanza

La función de semejanza seleccionada para medir la similaridad que existe entre un problema (P) y un caso (C) en la solución propuesta es:

$$\beta(P, C) = \sum_{i=1}^n w_i * \delta_i(P_i, C_i)$$

Donde w_i es la importancia del rasgo, P_i y C_i son los valores que el rasgo i tiene en el problema y en el caso respectivamente, y δ_i es la función de comparación para el rasgo i .

Importancia de los Rasgos (w_i):

Cada tema contará con rasgos predictores que representan las preguntas del tema. El peso de cada rasgo se definió según criterio de expertos a partir de la importancia que tiene el dominio de la pregunta por parte del estudiante, la suma de los pesos de las preguntas de cada tema es igual a 1.

Función de comparación

A partir de que los cuestionarios que presenta el sistema cuentan con preguntas de selección múltiple, surge la necesidad de utilizar una función de comparación entre conjuntos, estos conjuntos representan los incisos que contiene este tipo de preguntas. Entre las funciones de comparación de conjuntos se encuentra la similitud basada en el **coseno** (34), la cual muestra una buena medida de similitud entre dos vectores en un espacio. El cálculo de esta similitud está dado por el coseno del ángulo formado por la representación de estos dos vectores en el espacio n-dimensional, tal y como lo muestra la expresión:

$$sim_{ik} = \cos_{ik} = \frac{r_i \cdot r_k}{\|r_i\| * \|r_k\|} = \sum_j \frac{r_{ij}}{\sqrt{\sum_j r_{ij}^2}} \frac{r_{kj}}{\sqrt{\sum_j r_{kj}^2}}$$

Otra de las funciones utilizadas es el **coeficiente de correlación de Pearson**. Este método indica la correlación lineal de los elementos de un vector con los del otro, lo

que de manera gráfica se vería si se toma el plano cartesiano, en uno de los ejes se ubican los valores asociados a un vector; en el otro eje los asociados al otro vector, y se determina la cercanía a la existencia de una recta que contenga todos los puntos de intersección de los valores de un eje con los de otro (35). Este coeficiente está dado por la siguiente expresión:

$$sim(a, b) = \frac{\sum_z (V_{a,z} - V_a)(V_{b,z} - V_b)}{\sqrt{\sum_e (V_{a,z} - V_a)^2 \sum_z (V_{b,z} - V_b)^2}}$$

El coeficiente de **Jaccard-Tanimoto** es otra de las alternativas utilizadas para medir la similaridad entre dos conjuntos, basándose en la cantidad de elementos comunes que poseen; se define para los conjuntos a y b de la siguiente manera:

$$sim(a, b) = \frac{V_c}{V_a + V_b - V_c}$$

Siendo V_a la cantidad de elementos del conjunto a; V_b la cantidad de elementos del conjunto b y V_c la cantidad de elementos comunes de a y b (36).

Luego de realizar un estudio sobre estas funciones de comparación se concluye con que las funciones basadas en el **Coseno** y el **coeficiente de correlación de Pearson** son usualmente utilizadas como medida de similitud entre conjuntos que presentan valores cuantitativos. Sin embargo, en la propuesta de solución se hallará la similaridad entre conjuntos que presentan valores categóricos (1 si el inciso fue marcado o 0 si no fue seleccionado), y por esta razón la aplicación de Pearson o del Coseno no es adecuada. En estos casos es recomendable utilizar el coeficiente de **Jaccard-Tanimoto**.

2.3.2 Fase de adaptación

Luego de ser seleccionados los casos de la BC más semejantes al caso a resolver, se procede a desarrollar el método de adaptación. En la solución propuesta el método de adaptación es la adaptación nula, producto a que la solución (bibliografía) que

propone un problema similar al problema a resolver es utilizada para dar solución al nuevo caso.

2.3.3 Fase de almacenamiento

La fase almacenamiento es desarrollada por el sistema luego de darle solución a un nuevo caso, el mismo es almacenado a la BC con sus rasgos predictores y el objetivo, el cual representa la bibliografía correspondiente al caso similar que dio solución al mismo.

Si el caso no es resuelto porque no existe uno similar en la BC se le dará la opción al profesor de agregar el nuevo caso a la BC.

2.4 Personas relacionadas con el Sistema

La siguiente tabla muestra las personas relacionadas con el Modelo del profesor como parte de un Sistema Tutorial Inteligente para recomendar de forma diferenciada la bibliografía adecuada a los estudiantes en la asignatura Redes y Seguridad Informática.

Personas	Justificación
Estudiante	Persona que al autenticarse el sistema le brinda la posibilidad de realizar un cuestionario de un tema determinado, observar la calificación obtenida y la documentación necesaria para su preparación según sus necesidades.
Profesor	Persona que cuenta con todos los privilegios sin incluir la gestión de usuarios del sistema.
Administrador	Persona encargada de gestionar el sistema, gestiona todos los usuarios otorgándoles permisos y privilegios. Además tiene privilegios de gestionar la información publicada en el sistema.

Tabla 1. Personas relacionadas con el sistema.

2.5 Lista de reserva del producto

La metodología XP define que la Lista de reserva del producto (LRP) contiene los requisitos funcionales del sistema, los cuales están ordenados según la prioridad de implementación, además incluye los requisitos no funcionales del sistema.

Las funcionalidades específicas que presenta el Modelo del profesor como parte de un Sistema Tutorial Inteligente para recomendar de forma diferenciada la bibliografía adecuada a los estudiantes en la asignatura Redes y Seguridad Informática son:

RF1. Autenticar Usuario.

RF2. Mostrar Presentación

RF3. Gestionar Usuario.

RF4. Gestionar Cuestionarios.

RF5. Mostrar Cuestionarios

RF6. Gestionar Materiales Educativos.

RF7. Mostrar nota y recomendaciones.

RF8. Almacenar casos.

Requisitos no funcionales:

Seguridad:

- Confidencialidad: la información manejada por el sistema está protegida de acceso no autorizado.
- Integridad: el sistema debe garantizar que la información se mantenga inalterable durante todo el proceso de uso del mismo.
- Disponibilidad: a los usuarios autorizados se les garantizará el acceso a la información deseada en el momento dado.

Apariencia o interfaz externa:

El nombre del sistema debe estar ubicado en la parte superior izquierda del sitio. El color de los textos debe contrastar con el del fondo y el tamaño de fuente debe ser lo suficientemente grande para que la información contenida en la misma sea fácil de leer.

Software

- **Requerimientos mínimos para el cliente:** debe tener instalado navegador Mozilla Firefox 38.0.
- **Requerimientos mínimos para el servidor:** Se requiere la instalación del gestor de base de datos PostGreSQL 9.4.1, servidor web Apache 2.2.6, PHP 5.3.

Hardware

Los requerimientos mínimos para la ejecución de la aplicación se resumen en: Procesador a 2.0 GHz de velocidad de procesamiento o superior y 1 GB de RAM o superior y 5 GB de espacio libre en disco duro.

Usabilidad

La aplicación propuesta poseerá una interfaz amigable, sencilla e intuitiva al usuario que se relacionen con el sistema.

2.6 Fase de exploración

La fase de Exploración es la primera fase definida por la metodología XP. Esta fase comienza por la creación de una serie de Historias de Usuarios (HU), las cuales definen mediante su redacción, lo que realmente desea el cliente con el sistema. Es aquí donde se define el alcance real del sistema, permitiendo una familiarización del equipo de desarrollo con las herramientas y tecnologías a utilizar para el desarrollo del sistema.

2.6.1 Historias de usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Representan una breve descripción del comportamiento del sistema. Estas son escritas desde la perspectiva del cliente, donde describen de forma breve las características que el sistema debe realizar, aunque también los desarrolladores pueden brindar su ayuda en la identificación de las mismas. Se realiza una historia de usuario por cada característica principal del sistema. El contenido de las historias debe ser concreto, sencillo y flexible (18).

2.6.2 Clasificación de las HU

Las HU serán representadas mediante tablas divididas por las siguientes secciones:

- **Número:** Número de HU.
- **Nombre de Historia de Usuario:** El nombre de la historia de usuario sería para identificarlas mejor entre los desarrolladores y el cliente.
- **Usuario:** Personas involucradas con la HU.
- **Iteración Asignada:** Número de la iteración en la que será realizada la HU.
- **Prioridad en negocio:** Se le otorga una prioridad (Alta, Media, Baja) a las HU de acuerdo a la necesidad de desarrollo.
- **Nivel de Complejidad:** Se clasifican en (Alta, Media o Baja) según su complejidad.
- **Estimación:** Tiempo estimado que demorará el desarrollo de la HU.
- **Descripción:** Breve descripción de la HU.
- **Observaciones:** Describe al requisito que hace referencia la HU.

2.6.3 Historias de Usuarios del STI

A continuación se muestran 3 de las HU del sistema, el resto de las HU se pueden encontrar en el **Anexo1**.

HISTORIA DE USUARIO	
Nombre: Autenticar Usuario	No: 1
Usuario: Estudiante, Profesor y Administrador	Prioridad en el Negocio: Alta
Nivel de Complejidad: Media	Estimación: 1 semana
Iteración Asignada: 1	Observaciones: Se hace referencia al requisito funcional Autenticar Usuario.

Descripción: El usuario debe autenticarse para ser identificado. Si es el Profesor, tiene todos los privilegios en la aplicación excepto gestionar usuarios. Si está registrado como estudiante solo podrá realizar cuestionarios y ver las evaluaciones y bibliografías recomendadas por el sistema. En caso de ser el Administrador, tiene todos los privilegios en el sistema.

Tabla 2. Historia de usuario: Autenticar usuario.

HISTORIA DE USUARIO	
Nombre: Mostrar Cuestionarios	No: 5
Usuario: Profesor, Estudiante y Administrador.	Prioridad en el Negocio: Alta
Nivel de Complejidad: Media	Estimación: 5 días
Iteración Asignada: 2	Observaciones: Se hace referencia al requisito funcional Mostrar Cuestionarios.
Descripción: La presente historia de usuario debe permitirles al profesor, al administrador y al estudiante, ver todos los cuestionarios existentes en el sistema.	

Tabla 3. Historia de usuario: Mostrar cuestionario.

HISTORIA DE USUARIO	
Nombre: Mostrar nota y recomendaciones.	No: 7
Usuario: Estudiante	Prioridad en el Negocio: Alta
Nivel de Complejidad: Alta	Estimación: 3 semanas
Iteración Asignada: 3	Observaciones: Se hace referencia al requisito funcional Mostrar Materiales Educativos.

Descripción: La presente historia de usuario debe permitirle al estudiante ver la nota alcanzada y la bibliografía recomendada por el sistema luego de realizar un cuestionario.

Tabla 4. Historia de usuario: Mostrar materiales educativos.

2.7 Fase de Planificación

En esta fase el cliente establece la prioridad que presenta cada HU, estas son organizadas en iteraciones, definiendo con el cliente la fecha de entrega de las mismas, cuya entrega debe obtenerse en no más de tres meses.

2.7.1 Plan de Iteraciones

Tiempo en semanas	Orden de implementación de las HU	Iteración
3 semanas	Autenticar Usuario Mostrar Presentación Gestionar Usuario	1
3 semanas	Gestionar Cuestionarios Gestionar Materiales Educativos Mostrar Cuestionarios	2
3 semanas	Mostrar nota y recomendaciones	3
2 semanas	Decidir almacenamiento de un nuevo caso	4

Tabla 5. Plan de Iteraciones.

2.8 Plan de entregas

Sistema	Final 1ra Iteración(1ra semana de marzo)	Final 2da Iteración(4ta semana de marzo)	Final 3ra Iteración(3ra semana de abril)	Final 4ra Iteración(1ra semana de mayo)
STI	Versión 0.1	Versión 0.2	Versión 0.3	Versión 1.0

Tabla 6. *Plan de entregas.*

2.9 Conclusiones parciales

En este capítulo se describió, el impacto social del sistema, la propuesta de solución, se identificó la lista de reserva del producto, las fases de Exploración y Planificación pertenecientes a la primera etapa del ciclo de vida de todo tipo de proyecto que siga la metodología XP para su desarrollo, detallando las HU identificadas, el plan de iteraciones y el plan de entrega.

CAPÍTULO 3. Diseño, Implementación y Prueba.

3.1 Introducción

En este capítulo se define la arquitectura del sistema, los patrones de diseño utilizados, así como las Tarjetas Clase-Responsabilidad-Colaborador y las Tareas de ingeniería generadas por cada HU. Además se muestra el Diagrama de despliegue y de Modelo de datos de la aplicación. De igual modo se reflejan las pruebas realizadas y la validación de la propuesta de solución, garantizando el cumplimiento del objetivo trazado en el desarrollo del sistema.

3.2 Patrón de la arquitectura

Symfony es el framework que rige la arquitectura del STI; el mismo está basado en un patrón clásico del diseño web conocido como arquitectura Modelo–Vista-Controlador (MVC). Este patrón está formado por tres niveles: el Modelo, el cual representa la información con la que trabaja la aplicación, la Vista, la cual genera una representación visual al usuario de los datos representados en el Modelo y el Controlador que se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. La arquitectura MVC proporciona grandes ventajas como la organización del código, la reutilización y la flexibilidad (37).

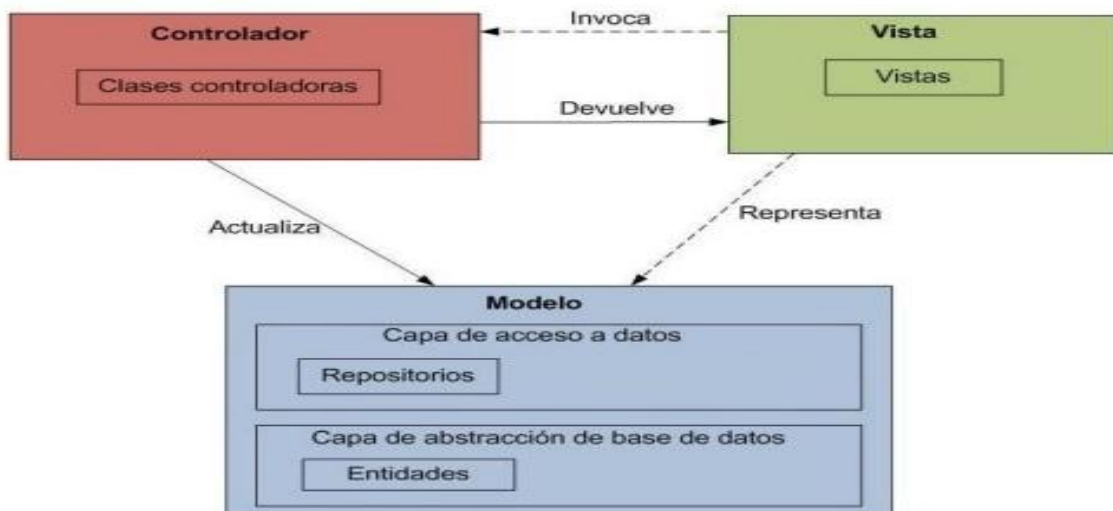


Figura 2: Arquitectura MVC del sistema.

Modelo

Esta capa se divide en Capa de acceso a datos, donde se encuentran los repositorios y Capa de abstracción de base de datos, donde se localizan las entidades.

En el Modelo se encuentra el *Object Relational Mapper* (ORM) Doctrine para el mapeo de objetos a la base de datos.

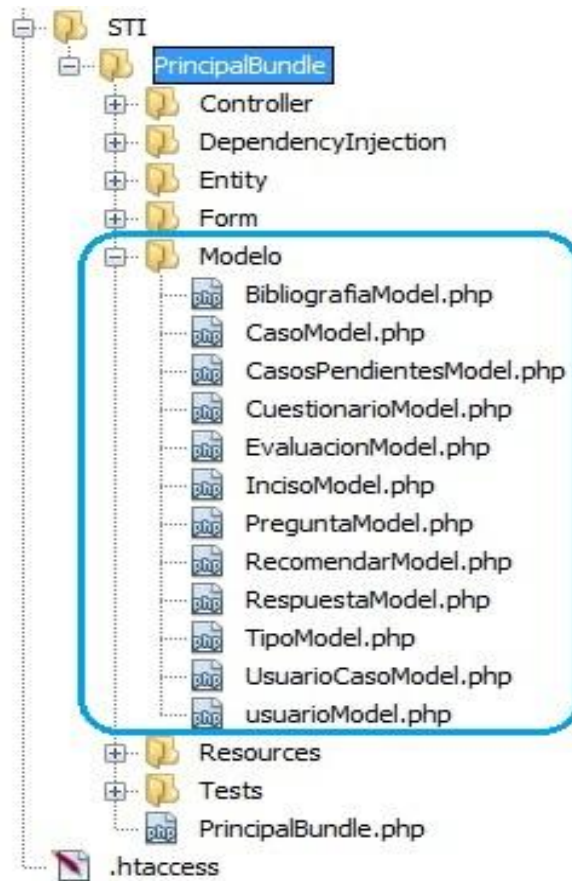


Figura 3: Clases Repositorios de la Capa de acceso a datos.

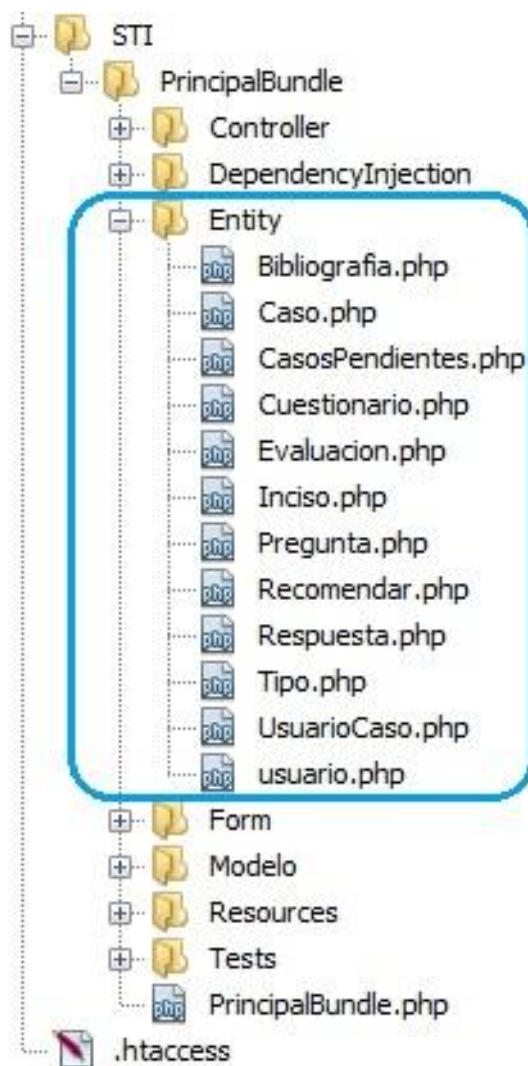


Figura 4: Entidades de la Capa de abstracción de base de datos.

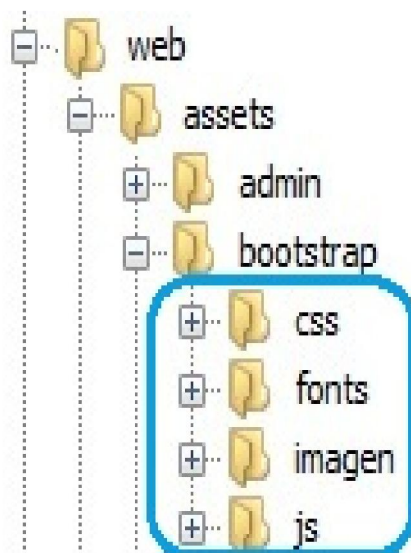
Vista

El sistema está compuesto por una plantilla base que se encuentra en el paquete `sti\app\Resources\views`:



Figura 5: Plantilla base.

Esta se encarga de importar los archivos que se encuentran en el paquete `sti\web\assets\bootstrap`:



El sistema además contiene una plantilla inicio que hereda de base, esta plantilla está compuesta por 2 layout, el del menú que nunca varía y el del contenido que puede variar en dependencia a la opción seleccionada en el menú.

Las clases que responden a la Vista se localizan en el paquete `sti\src\STI\PrincipalBundle\Resources\views` y heredan de la plantilla Inicio, las mismas son mostradas a continuación.

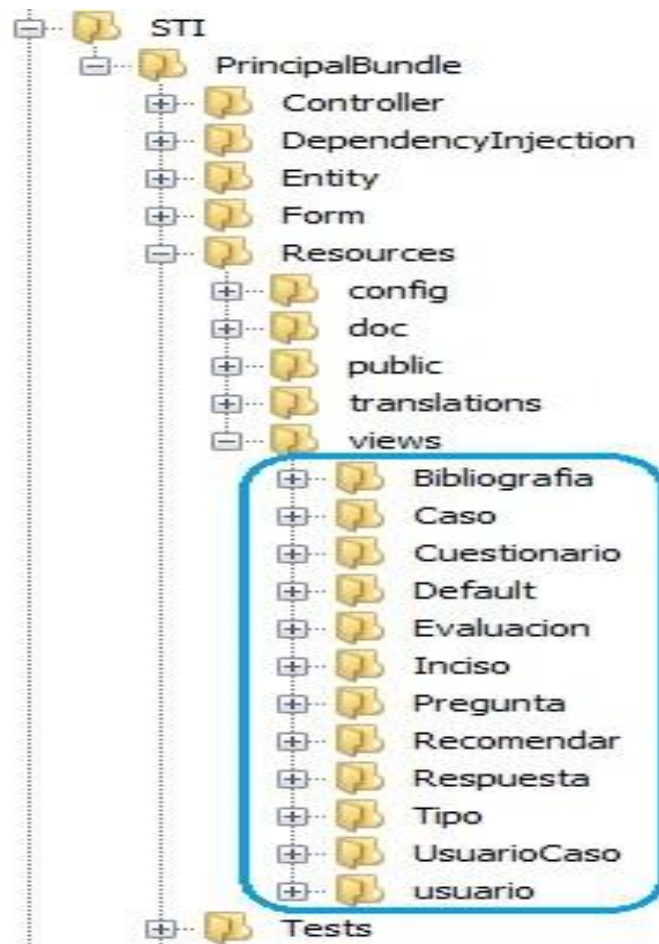


Figura 6: Vistas del STI.

Controlador

El controlador es la piedra angular de la arquitectura MVC, permite enlazar el modelo con la vista y es básicamente donde se construye la aplicación. En esta capa se encuentran las clases controladoras, las cuales se localizan dentro del paquete `sti\src\STI\PrincipalBundle\Controller` y se muestran a continuación.

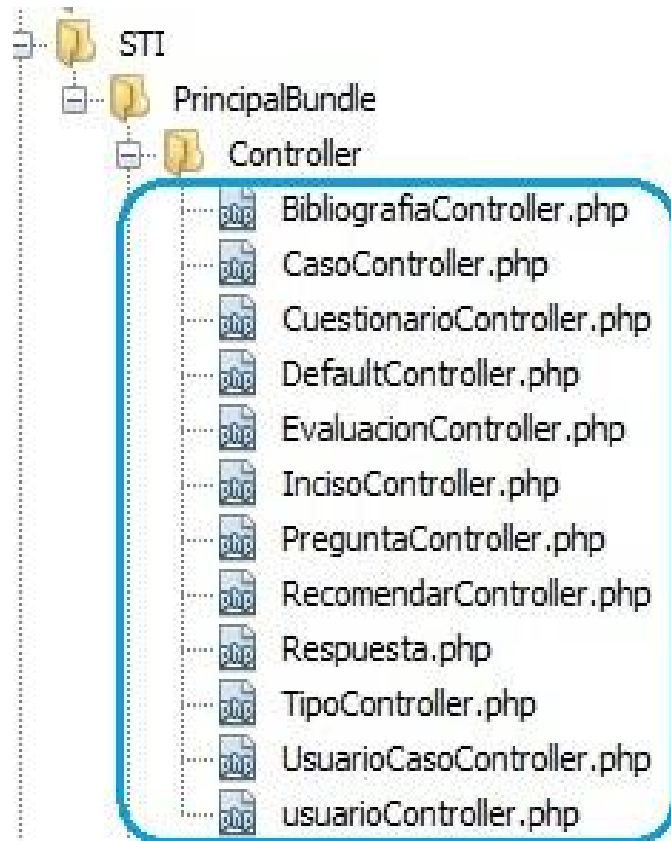


Figura 7: Clases controladoras.

3.3 Patrones de diseño

Los patrones de diseño proveen a los equipos de desarrollo de un mecanismo que les permitirá diseñar aplicaciones de alta calidad en el menor tiempo posible, debido a que constituyen soluciones a problemas específicos del diseño, promueven la reutilización del código y la agilización del proceso de desarrollo de software.

Los patrones de diseño expresan esquemas para definir estructuras de diseño para el desarrollo de software. En el diseño del STI se utilizan los Patrones Generales de Software para Asignar Responsabilidades (GRASP). Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Dentro de los patrones GRASP utilizados para el desarrollo del STI se encuentran:

- El patrón **Experto** es el encargado de asignar una responsabilidad al experto en información, el cual no es más que la clase que cuenta con la información necesaria para cumplir la responsabilidad. En la implementación del STI el uso de este patrón

se evidencia por ejemplo, cuando es necesario renderizar las vistas para mostrarlas al usuario. En este caso las clases controladoras son las expertas en información y crean los formularios que permiten mostrar los campos requeridos en la vista.

```
public function newAction() {
    $entity = new Cuestionario();
    $em = $this->getDoctrine()->getManager();
    $form = $this->createForm(new CuestionarioType(), $entity);
    $cuestionarios = $em->getRepository('PrincipalBundle:Cuestionario')->findAll();
    return $this->render('PrincipalBundle:Cuestionario:new.html.twig', array(
        'entity' => $entity,
        'form' => $form->createView(),
        'cuestionarios' => $cuestionarios
    ));
}
```

Figura 8: Ejemplo del patrón Experto en el STI.

- El patrón **Creador** es el responsable de asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A. En el desarrollo del STI, la implementación de este patrón se evidencia cuando las clases controladoras crean los formularios ya que son las responsables de crear instancias de la clase Type, también cuando la clase controladora crea las entidades que se van a asociar a un determinado formulario.

```
public function newAction() {
    $entity = new Cuestionario();
    $em = $this->getDoctrine()->getManager();
    $form = $this->createForm(new CuestionarioType(), $entity);
    $cuestionarios = $em->getRepository('PrincipalBundle:Cuestionario')->findAll();
    return $this->render('PrincipalBundle:Cuestionario:new.html.twig', array(
        'entity' => $entity,
        'form' => $form->createView(),
        'cuestionarios' => $cuestionarios
    ));
}
```

Figura 9: Ejemplo del patrón Creador en el STI.

- El patrón **Bajo Acoplamiento** es el encargado de asignar una responsabilidad para mantener bajo acoplamiento con el objetivo de tener las clases lo menos

ligadas entre sí, de forma que si se realiza modificación en alguna de ellas, tenga la mínima repercusión en el resto de las clases.

En el desarrollo del STI, este patrón se evidencia en la Capa del modelo, donde las clases de abstracción de base de datos (Entidades) que son las más reutilizables, no tienen asociaciones con las clases de la Capa de la vista ni con la Capa del controlador.

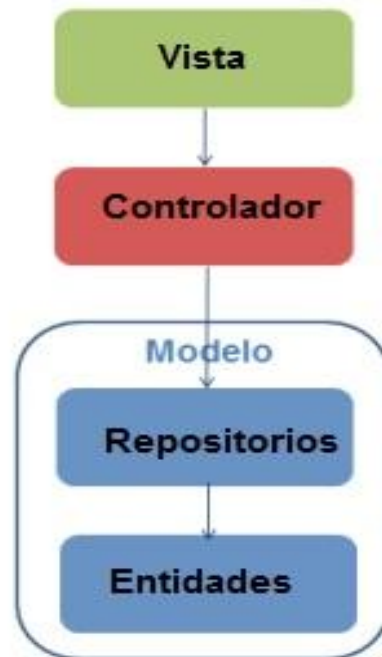


Figura 10: Ejemplo del patrón Bajo Acoplamiento en el STI.

El patrón **Alta Cohesión** tiene como objetivo garantizar que cada una de las clases tenga responsabilidades específicas y colabore con otras. En el desarrollo del STI, este patrón se evidencia cuando las clases del modelo contienen métodos que realizan consultas a la Base de Datos y son usados en las clases controladoras.

Método `findByPregunta` de la clase `PreguntaModel`:

```
public function findByPregunta($idcuestionario) {  
    return $this->getEntityManager()  
        ->createQuery(  
            'SELECT p FROM PrincipalBundle:Pregunta p WHERE'  
            . ' p.idCuestionario=:cuest ORDER BY p.id ASC'  
        )  
        ->setParameter('cuest', $idcuestionario)  
        ->getResult();  
}
```

En la clase `RespuestaController` se llama al método `finByPregunta` de la clase `PreguntaModel`:

```
$preguntas = $em->getRepository('PrincipalBundle:Pregunta')->findByPregunta($idcuestionario);
```

Figura 11: Ejemplo del patrón Alta Cohesión en el STI

3.4 Tarjetas Clase-Responsabilidad-Colaborador

La metodología XP sugiere diseños simples y sencillos, para ello propone utilizar las Tarjetas Clase-Responsabilidad-Colaborador, las cuales son empleadas con el objetivo de organizar de forma simple, las clases más relevantes para las funcionalidades del sistema.

A continuación se muestran las Tarjetas Clase-Responsabilidad-Colaborador correspondientes a algunas de las clases controladoras del STI.

Tarjeta CRC clase Caso	
Super Classes: CasoControlller	
Descripción: Clase encargada de controlar la gestión de los casos en el sistema.	
Atributes:	
Nombre	Descripción
Caso_Model	Permite el acceso a los datos de la entidad Caso
Responsabilidades:	
Nombre	Collaborator
Gestionar Casos	Caso_Model

Figura 12: Tarjeta CRC de la clase: CasoController.

Tarjeta CRC clase Cuestionario	
Super Classes: CuestionarioController	
Descripción: Clase encargada de controlar la gestión de los casos en el sistema.	
Atributes:	
Nombre	Descripción
Cuestionario_Model	Permite el acceso a los datos de la entidad Cuestionario
Responsabilidades:	
Nombre	Collaborator
Gestionar Cuestionario	Cuestionario_Model

Figura 13: Tarjeta CRC de la clase: CuestionarioController.

Tarjeta CRC clase Pregunta	
Super Classes: PreguntaControllier	
Descripción: Clase encargada de controlar la gestión de las preguntas correspondientes a cada cuestionario en el sistema.	
Atributos:	
Nombre	Descripción
Pregunta_Model	Permite el acceso a los datos de la entidad Pregunta
Responsabilidades:	
Nombre	Collaborator
Gestionar Preguntas	Pregunta_Model

Figura 14: Tarjeta CRC de la clase: PreguntaController.

Tarjeta CRC clase Inciso	
Super Classes: IncisoControllier	
Descripción: Clase encargada de controlar la gestión de los incisos correspondientes a cada pregunta de los cuestionarios del sistema.	
Atributos:	
Nombre	Descripción
Inciso_Model	Permite el acceso a los datos de la entidad Inciso
Responsabilidades:	
Nombre	Collaborator
Gestionar Incisos	Inciso_Model

Figura 15: Tarjeta CRC de la clase: IncisoController.

3.5 Modelo de datos

Un modelo de datos es una representación abstracta de los datos de un sistema y las relaciones existentes entre ellos. El diseño de la base de datos es fundamental para el desarrollo de cualquier aplicación. A continuación se muestra el modelo físico de la base de datos para la propuesta de solución.

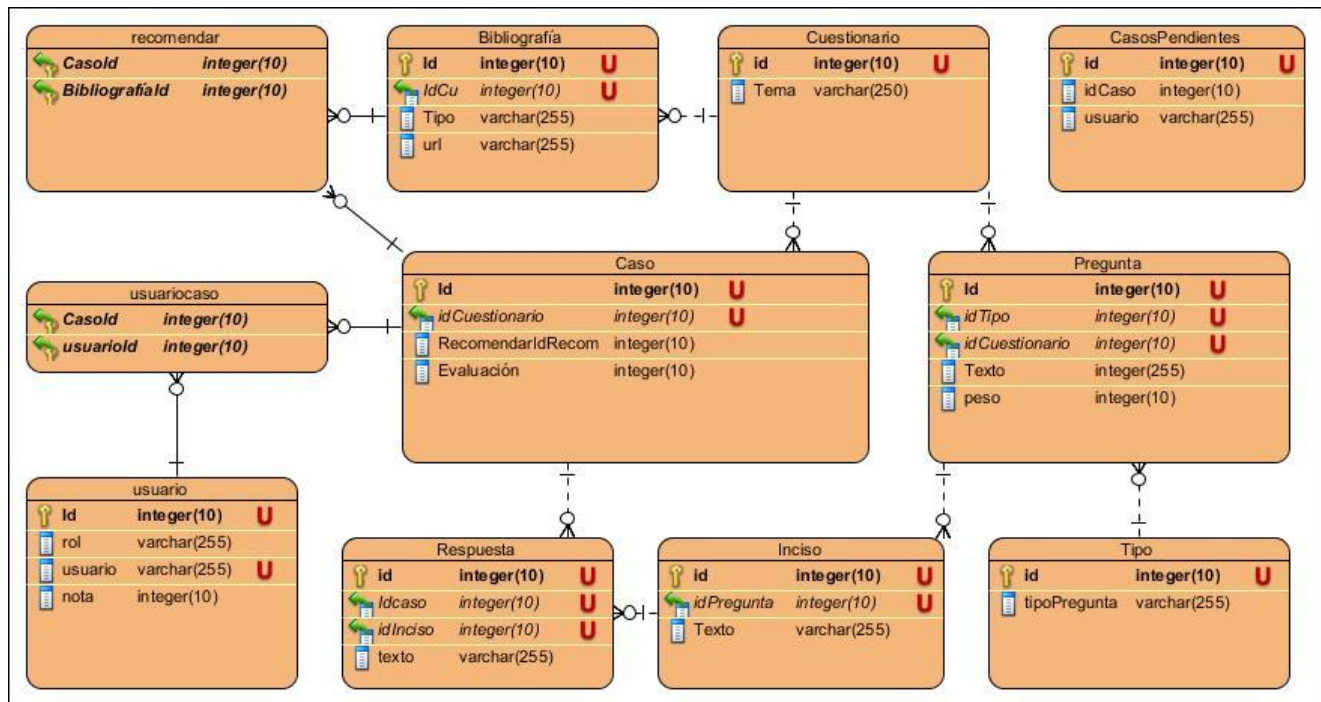


Figura 16: Modelo de datos.

3.6 Diagrama de despliegue

El diagrama de despliegue es un tipo de diagrama UML que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones. Muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Un nodo es un recurso de ejecución como un computador, dispositivo, conectados por asociaciones de comunicación tales como enlaces de red, conexiones HTTP.



Figura 17: Diagrama de despliegue.

Descripción de los componentes:

PC_Cliente: El PC_Cliente es donde se podrá mostrar e interactuar con el sistema, a través de un navegador web previamente instalado en el ordenador.

Apache: El Servidor Apache es donde se encuentra alojada la aplicación web, a la que se conectan todos los clientes por medio de sus ordenadores.

PostgreSQL: En el Servidor PostgreSQL es donde se encuentran almacenados todos los datos que permanecen en el sistema.

HTTP: El protocolo HTTP se utiliza para conectar el ordenador del cliente con el servidor web.

TCP/IP: TCP/IP es el protocolo de comunicación entre el servidor web y el servidor de base de datos empleado, este proporciona la transferencia confiable de paquetes de datos sobre la red.

3.7 Tareas de ingeniería

La metodología XP plantea que durante la fase de implementación, se desarrollen las HU en sus respectivas iteraciones, dividiendo las mismas en tareas de ingeniería que serán implementadas por los desarrolladores, las cuales pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

Iteración 1

El principal objetivo de esta iteración es desarrollar las HU:

- Autenticar usuario
- Mostrar presentación

- Gestionar usuario

Para la HU Autenticar usuario son desarrolladas las siguientes tareas:

Tarea N° 1: Comprobar existencia del usuario en LDAP y rol que ocupa en la base de datos.

Tarea N° 2: Mostrar sección correspondiente y guardar datos en la base de datos.

Para la HU Mostrar presentación son desarrolladas las siguientes tareas:

Tarea N° 3: Mostrar significado y objetivos generales del STI, así como noticias de los temas impartidos en la asignatura Redes y Seguridad Informática.

Tarea N° 4: Salir del sistema en cualquier momento.

Para la HU Gestionar usuario son desarrolladas las siguientes tareas:

Tarea N° 5: Insertar usuario

Tarea N° 6: Listar usuarios

Tarea N° 7: Modificar usuario

Tarea N° 8: Eliminar usuario

Iteración 2

El principal objetivo de esta iteración es desarrollar las HU:

- Gestionar Cuestionarios
- Mostrar Cuestionarios
- Gestionar Materiales Educativos

Para la HU Gestionar cuestionarios se desarrollan las siguientes tareas:

Tarea N° 9: Insertar cuestionarios

Tarea N° 10: Modificar cuestionario

Tarea N° 11: Eliminar cuestionario

Para la HU Mostrar Cuestionarios se desarrollan las siguientes tareas:

Tarea N° 12: Mostrar el cuestionario correspondiente a cada tema de la asignatura para ser desarrollados.

Para la HU Gestionar Materiales Educativos se desarrolla la siguiente tarea:

Tarea N° 13: Insertar materiales educativos.

Tarea N° 14: Eliminar materiales educativos.

Iteración 3

El principal objetivo de esta iteración es desarrollar la HU:

Mostrar nota y recomendaciones.

Para la HU Mostrar nota y recomendaciones se desarrollan las siguientes tareas:

Tarea N° 15: Aplicar función de semejanza al nuevo caso con cada uno de los casos que se encuentran en la BC.

Tarea N° 16: Determinar caso más semejante teniendo en cuenta el umbral de semejanza y mostrarle al usuario la nota y la bibliografía asociada a dicho caso.

Iteración 4

El principal objetivo de esta iteración es desarrollar la HU:

Decidir almacenamiento del nuevo caso.

Para la HU Decidir almacenamiento del nuevo caso se desarrollan las siguientes tareas:

Tarea N° 17: Mostrar posibles casos a agregar a la BC.

Tarea N° 18: Insertar un nuevo caso.

Tarea N° 19: Eliminar los casos.

Las tareas de la ingeniería serán representadas mediante tablas que contienen los siguientes aspectos:

Nombre de la Tarea: Representa el nombre de la tarea.

Número de Tarea: Representa el número de la tarea.

Número de HU: Representa el número de la HU a la que pertenece dicha tarea.

Estimación: Representa el tiempo estimado para el desarrollo de la tarea.

Fecha Inicio: Representa la fecha en que inicia el desarrollo de la tarea.

Fecha Fin: Representa la fecha en que concluye el desarrollo de la tarea.

Tipo de Tarea: (Desarrollo, Corrección, Mejora).

Programador Responsable: Representa el nombre y los apellidos del programador responsable de desarrollar la tarea.

Descripción: Breve descripción de la tarea.

A continuación se presentan algunas de las Tareas de ingeniería del sistema, el resto se pueden encontrar en el **Anexo2**.

Tarea de Ingeniería	
Nombre de la Tarea: Comprobar existencia del usuario en LDAP y rol que ocupa en la base de datos.	Número de Tarea: 1
Número de HU: 1	Estimación: 2 días
Fecha Inicio: 10/2/2015	Fecha Fin: 12/2/2015
Tipo de Tarea: Desarrollo	Programador Responsable: Alejandro Mojena Elías y José Alejandro Perez Rodriguez

Descripción: El sistema comprueba la existencia del usuario en LDAP y el rol que ocupa en la base de datos para mostrar la sección correspondiente.

Tabla 7. Tarea de ingeniería: Comprobar existencia del usuario en LDAP y rol que ocupa en la base de datos.

Tarea de Ingeniería	
Nombre de la Tarea: Mostrar sección correspondiente y guardar datos en la base de datos.	Número de Tarea: 2
Número de HU: 1	Estimación: 3 días
Fecha Inicio: 12/2/2015	Fecha Fin: 15/2/2015
Tipo de Tarea: Desarrollo	Programador Responsable: Alejandro Mojena Elías y José Alejandro Perez Rodriguez
Descripción: El sistema muestra una sección según el rol que ocupa el usuario autenticado.	

Tabla 8. Tarea de ingeniería: Mostrar sección correspondiente y guardar datos de la base de datos.

3.8 Pruebas

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente, todo esto contribuye a elevar la calidad de los sistemas desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones. Existen 2 maneras de probar un sistema: las pruebas de caja blanca y las pruebas de caja negra (38).

Las pruebas de caja blanca es un método que utiliza la estructura de control descrita como parte del diseño al nivel de componentes para derivar los casos de pruebas. Una de las técnicas de prueba de caja blanca es la ruta básica. Este método de ruta básica permite que el diseñador de casos de pruebas obtenga una medida de complejidad lógica de un diseño procedimental y que use esta medida como guía un conjunto

básico de rutas de ejecución. Los casos de prueba derivados para ejercitar el conjunto básico deben garantizar que se ejecuten cada instrucción del programa por lo menos una vez durante la prueba (38).

Los pasos que se siguen para aplicar esta técnica son:

- A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
- Se calcula la complejidad ciclomática del grafo.
- Se determina un conjunto básico de caminos independientes.
- Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico (38).

Técnica de ruta básica	Mostrar nota y recomendaciones
Probadores: Jose Alejandro Perez Rodriguez y Alejandro Mojena Elías	
<p>Código al que se aplica:</p> <pre> public function mostrar_recomendadaAction(\$id) { \$em = \$this->getDoctrine()->getManager(); \$cuestionario = \$em->getRepository('PrincipalBundle:Cuestionario')->find(\$id); \$cuestionarios = \$em->getRepository('PrincipalBundle:Cuestionario')->findAll(); \$usuario = \$this->get('session')->get('usuario'); \$usuario = \$em->getRepository('PrincipalBundle:usuario')->findOneBy(array('usuario' => \$usuario)); \$usuariocaso = \$em->getRepository('PrincipalBundle:UsuarioCaso')->findBy(array('idUserario' => \$usuario->getId())); \$casospendientes = \$em->getRepository('PrincipalBundle:CasosPendientes')->findBy(array('Usuario' => \$usuario->getUsuario())); \$texto2 = "Antes de ver la bibliografía recomendada del tema: " . \$cuestionario->getTema() . " debe responder el cuestionario de dicho tema "; if (count(\$usuariocaso) == 0) { if (count(\$casospendientes) == 0) { \$this->get('session')->getFlashBag()->add('info', \$texto2); } } } </pre>	

```

    );

    return $this->render('PrincipalBundle:Cuestionario:Temas.html.twig', array(
        'cuestionarios' => $cuestionarios,
        'id' => $id
    ));
} else {
    $this->get('session')->getFlashBag()->add(
        'info', 'Debe esperar que el profesor le asigne la bibliografía recomendada'
    );

    return $this->render('PrincipalBundle:Cuestionario:Temas.html.twig', array(
        'cuestionarios' => $cuestionarios,
        'id' => $id
    ));
}
} else {
    foreach ($usuariocaso as $uc) {
        $idcaso = $uc->getIdCaso()->getId();

        if ($uc->getIdCaso()->getIdCuestionario()->getId() == $id) {
            $bibliografia = $em->getRepository('PrincipalBundle:Recomendar')->findBy(array('idCaso' => $uc->getIdCaso()->getId()));
        }
    }

    $caso = $em->getRepository('PrincipalBundle:Caso')->find($idcaso);

    $evaluacion = $caso->getEvaluacion();

    $texto3 = "luego de realizar el cuestionario del tema: ". $cuestionario->getTema() . "y alcanzar una nota de " .
    $evaluacion . " puntos se considera que no necesita de una bibliografía específica para";

    if ($bibliografia == null) {
        return $this->render('PrincipalBundle:Bibliografia:recomendada.html.twig', array(
            'bibliografia' => $BibliografiaRecomendada,
            'cuestionario' => $cuestionario,

```

```
'cuestionarios' => $cuestionarios,

'texto' => $texto3,

));

}

$texto1 = "luego de realizar el cuestionario del tema: ". $cuestionario->getTema() . "y alcanzar una nota de " .
$evaluacion . " puntos se le sugiere que se apoye en la siguiente bibliografía:";

$bibliografiarecomendada = array();

foreach ($bibliografia as $b) {

    if ($b->getIdBibliografia() != NULL) {

        $bibliografiarecomendada[] = $em->getRepository('PrincipalBundle:Bibliografia')->
find($b->getIdBibliografia()->getId());

    } else {

        $BibliografiaRecomendada = array();

    }

}

return $this->render('PrincipalBundle:Bibliografia:recomendada.html.twig', array(

    'bibliografia' => $bibliografiarecomendada,

    'cuestionario' => $cuestionario,

    'cuestionarios' => $cuestionarios,

    'texto' => $texto1,

    'evaluacion' => $evaluacion,

));

}

}
```

Complejidad ciclomática:

1. $V(G) = (A - N) + 2.$
2. $V(G) = NP + 1.$
3. $V(G) = R + 1.$

Dónde: $\left\{ \begin{array}{l} A: \text{aristas.} \\ R: \text{regiones cerradas.} \\ N: \text{nodos.} \\ NP: \text{nodos predicados.} \end{array} \right.$

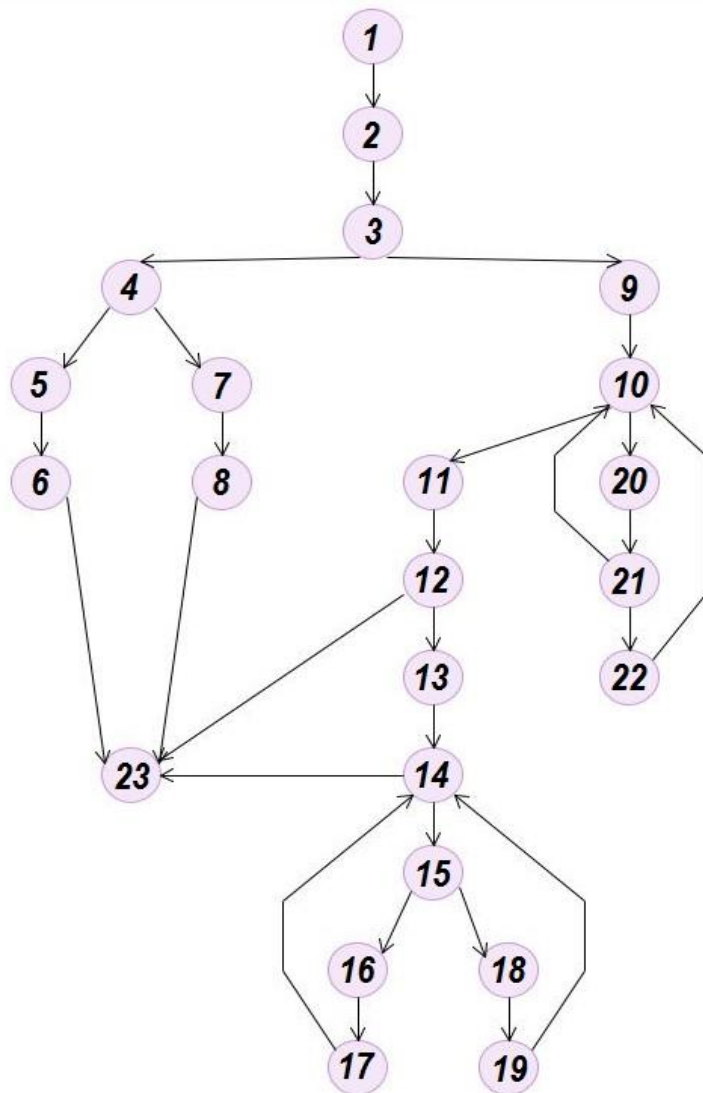
Sustituyendo los valores se obtiene que:

1. $V(G) = (29 - 23) + 2 = 8$
2. $V(G) = 7 + 1 = 8$
3. $V(G) = 7 + 1 = 8$

Caminos independientes:

- 1) 1-2-3-4-5-6-23
- 2) 1-2-3-4-7-8-23
- 3) 1-2-3-9-10-11-12-23
- 4) 1-2-3-9-10-11-12-13-14-23
- 5) 1-2-3-9-10-11-12-13-14-15-16-17-14-23
- 6) 1-2-3-9-10-11-12-13-14-15-18-19-14-23
- 7) 1-2-3-9-10-20-21-22-10-11-12-23
- 8) 1-2-3-9-10-20-21-22-10-11-12-13-14-23

Representación del grafo:



Caso de prueba para el camino básico 1

Descripción: El usuario desea acceder a la opción (Nota y Recomendaciones) de un tema sin haber realizado el cuestionario de dicho tema.

Condición de ejecución: El usuario accede a la opción (Nota y Recomendaciones) sin haber realizado el cuestionario.

Datos de entrada:

Id de un tema (x).

Resultado esperado:	Antes de ver la bibliografía recomendada del tema (x) debe responder el cuestionario de dicho tema.
Evaluación del caso de prueba :	Satisfactorio

Tabla 9: Caso de prueba para el camino básico 1.

Caso de prueba para el camino básico 8	
Descripción: El usuario desea acceder a la opción (Nota y Recomendaciones) de un tema luego de haber realizado el cuestionario de dicho tema y no haber obtenido la máxima calificación (5).	
Condición de ejecución: El usuario accede a la opción (Nota y Recomendaciones) luego de haber realizado el cuestionario y obtener una nota de 2, 3 o 4.	
Datos de entrada:	Id de un tema (x).
Resultado esperado:	Plantilla twig: refleja mensaje y lista de bibliografías. Mensaje = “Luego de realizar el cuestionario del tema (x) y alcanzar una nota de [2, 3, 4] puntos se le sugiere que se apoye en la siguiente bibliografía:” Listado de bibliografías.
Evaluación del caso de prueba :	Satisfactorio

Tabla 10: Caso de prueba para el camino básico 8.

Resultados de la técnica de prueba ruta básica

Se realizaron cuatro iteraciones de prueba de ruta básica y se diseñaron casos de prueba para cada uno de los caminos independientes que arrojaron las pruebas de caja blanca, identificando un total de 14 No Conformidades (NC). En la primera iteración se detectaron 8 NC, de ellas seis de lógica algorítmica y dos relacionadas

con mensajes de información incorrectos a los usuarios. En la segunda iteración se detectaron 4 NC, de ellas tres de lógica algorítmica y una de mensaje de información incorrecto a los usuarios. En la tercera iteración se detectaron 2 NC, ambas de lógica algorítmica. En la cuarta iteración se verificó que todas las NC encontradas con anterioridad se había resuelto satisfactoriamente y no se detectaron nuevas NC.

En la siguiente figura se muestra el número de iteraciones realizadas y la cantidad de NC detectadas en cada iteración. Estas pruebas permitieron ejecutar al menos en una ocasión cada sentencia del programa obteniendo resultados satisfactorios.



Figura 18: Resultado de la técnica de prueba ruta básica.

Las pruebas de aceptación son consideradas como pruebas de caja negra, son creadas en base a las HU, dichas historias no se pueden considerar terminadas hasta tanto pase correctamente por las pruebas de aceptación. Se debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada (39). Estas pruebas fueron realizadas al STI con el objetivo de verificar que el software está listo y que puede ser usado por los usuarios.

Estas pruebas son representadas mediante tablas divididas por los siguientes aspectos:

Código: código que identifica el caso de prueba.

Historia de Usuario: número de la HU que hace referencia el caso de prueba.

Nombre: nombre de la funcionalidad a la que se le hace la prueba.

Descripción: breve descripción del caso de prueba.

Condiciones de Ejecución: condiciones necesarias para ejecutar la prueba.

Entrada/ Pasos de ejecución: valores de entrada.

Resultado Esperado: salida de la ejecución.

Evaluación de la Prueba: evaluación de la prueba.

A continuación se presentan casos de prueba realizados a algunas de las HU del sistema, los casos de pruebas restantes se pueden encontrar en el **Anexo 3**.

Caso de prueba de aceptación
Código: C-P-A-1
Historia de Usuario: 1
Nombre: Autenticar usuario.
Descripción: Prueba para la funcionalidad Autenticar usuario.
Condiciones de Ejecución: El usuario debe introducir usuario y contraseña.
Pasos de ejecución: El usuario introduce usuario y contraseña y oprime el botón Ingresar. El sistema comprueba su existencia en LDAP y el rol que ocupa en la base de datos para mostrar la sección correspondiente al usuario autenticado.
Resultado Esperado: El usuario es identificado por el sistema y permite ver la interfaz correspondiente al rol que ocupa en la base de datos.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 11: Caso de prueba para la HU: Autenticar usuario.

Caso de prueba de aceptación
Código: C-P-A-2
Historia de Usuario: 5

Nombre: Mostrar cuestionario
Descripción: Prueba para la funcionalidad Mostrar cuestionario.
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema y debe seleccionar un tema de la asignatura.
Entrada/ Pasos de ejecución: El usuario luego de autenticarse y seleccionar el tema decide desarrollar el cuestionario de dicho tema seleccionando la opción “Cuestionario”.
Resultado Esperado: El sistema debe mostrar el cuestionario del tema seleccionado por el estudiante.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 12: Caso de prueba para la HU: Mostrar cuestionario.

Caso de prueba de aceptación
Código: C-P-A-3
Historia de Usuario: 7
Nombre: Mostrar nota y recomendaciones
Descripción: Prueba para la funcionalidad Mostrar nota y recomendaciones.
Condiciones de Ejecución: El usuario ha desarrollado un cuestionario de un determinado tema de la asignatura.
Entrada/ Pasos de ejecución: El usuario al realizar un cuestionario, el sistema le mostrará la nota alcanzada y le recomendará el estudio de un determinado material educativo si es necesario.
Resultado Esperado: El sistema debe mostrar el material educativo que corresponda a su nivel de conocimiento.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 13: Caso de prueba para la HU: Mostrar Nota y Recomendaciones.

Además del desarrollo de estos casos de pruebas se realizó una prueba de aceptación por parte del cliente para verificar que el sistema brinda una bibliografía adecuada a partir del nivel de conocimiento demostrado por un estudiante al realizar un cuestionario en el sistema. Se tomó una muestra de 20 estudiantes los cuales dieron solución a los cuestionarios existentes y en todos los casos el sistema recomendó una bibliografía acorde con sus necesidades, esto fue comprobado por dos expertos en temas relacionados con la asignatura Redes y Seguridad Informática.

3.9 Conclusiones parciales

En este capítulo se abordó sobre la etapa de diseño, implementación y prueba del Modelo del profesor como parte de un Sistema Tutorial Inteligente para recomendar de forma diferenciada la bibliografía adecuada a los estudiantes en la asignatura Redes y Seguridad Informática, describiendo la arquitectura utilizada para el desarrollo del sistema, se reflejaron los patrones de diseños utilizados, así como las tareas de ingeniería necesarias para cada HU, se presentó el diagrama de despliegue y de modelo de datos y se mostraron las pruebas realizadas al sistema.

CONCLUSIONES GENERALES

Con el desarrollo del presente trabajo se arriba a las siguientes conclusiones:

- El estudio realizado durante la investigación posibilitó analizar de forma detallada las principales características que presentan los STI existentes, así como de los SBC, adoptando el paradigma de creación de estos para el desarrollo de la solución propuesta.
- Se desarrolló un Modelo del profesor como parte de un Sistema Tutorial Inteligente con vista a recomendar la bibliografía adecuada a partir del nivel de conocimiento de los estudiantes para la auto-preparación de los mismos.
- Las pruebas realizadas al sistema permitieron demostrar que la aplicación cumple satisfactoriamente con los requisitos que garantizan su correcto funcionamiento.

RECOMENDACIONES

Luego de culminar el desarrollo del presente trabajo, se recomienda para futuras versiones:

- Garantizar que la base de casos sea capaz de determinar la incorporación de nuevos casos sin consultar con el experto humano, logrando así el auto-aprendizaje de la misma.
- Agregar otros tipos de preguntas a los cuestionarios para enriquecer la obtención del nivel de conocimiento de los usuarios.

Referencias

1. **Vidal Ledo, María y Gómez Martínez, Freddy.** *Software educativo.* Habana, Cuba : s.n., 2010.
2. **Martínez Sánchez, Natalia, García Lorenzo, María Matilde y García Valdivia, Zoila Zenaida.** *Sistemas Basados en Casos & Sistemas de Enseñanza-Aprendizaje Inteligentes.* Las Villas,Cuba : s.n., 2010.
3. **Parra, Franklin.** *Sistema Tutorial Inteligente.* Escuela Superior Politécnica del Litoral,Guayaquil, Ecuador : s.n., 2010.
4. **García Lorenzo, María Matilde, Martínez Sánchez, Natalia y García Valdivia, Zoila Zenaida.** *Modelo para diseñar sistemas de enseñanza-aprendizaje inteligentes utilizando el razonamiento basado en casos.* 2010.
5. **Linamientos de la Política Económica y Social del Partido y la Revolución.** Habana, Cuba : s.n., 2011.
6. **Sánchez, Natalia Martínez.** *Guía de orientación para la ingeniería del conocimiento en el diseño de sistemas de enseñanza-aprendizaje inteligentes utilizando el razonamiento basado en casos.* 2011.
7. **Morales Rodríguez, Luisa Idorka y Fernández, Yarisleydi de Avila .** *Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos.* La Habana : s.n., 2012.
8. **Nápoles Mejías, Edney y Rivera Peguero, Rosalba.** *Sistema para el montaje de un tutorial inteligente.* La Habana : s.n., 2013.
9. **Jiménez, Rey.** *Una Aplicación de la Tecnología de MultiAgentes a los Sistemas Tutores Inteligentes: Enseñanza de Computación en Carreras de Ingeniería.* Argentina : s.n., 2006.
10. **Cataldi, Zulma.** *Sistemas Tutores Inteligentes Orientados a la Enseñanza para la Comprensión.* Facultad Regional Buenos Aires. Argentina. : s.n., 2009.
11. **Ferreira, Anita, y otros.** *La Arquitectura de ELE-TUTOR: Un Sistema Tutorial Inteligente para el Español como Lengua Extranjera.* Universidad de Concepción, Chile : s.n., 2012.
12. **Morales Rodríguez, Luisa Idorka y Fernández, Yarisleydi .** *Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos.* Habana : s.n., 2012.
13. **Nápoles Mejías, Edney y Rivera Peguero, Rosalba.** *Sistema para el montaje de un tutorial inteligente.* Habana : s.n., 2013.
14. **Gil, Dr.María del Pilar Gómez.** *REDES NEURONALES ARTIFICIALES INAOE.* México : s.n., 2012.
15. **Aspiazu, Guillermo Choque.** *Sistemas expertos conexionistas.* 2010.
16. **Francisco J. Arias S., Jovani A. Jiménez B., Demetrio A. Ovalle C.** *Modelo de planificación instruccional en sistemas tutoriales inteligente.* Colombia : s.n., 2010. núm. 1.
17. **Soria Francis, Sindy.** *Modelo para diseñar Mapas Conceptuales Inteligentes utilizando el Razonamiento Basado en Casos.* Habana : s.n., 2010.
18. **Gimson, Loraine.** *Metodologías ágiles y desarrollo basado en conocimiento.* Buenos Aires, Argentina : s.n., 2012.

19. Laura Carabalí, Daniela Lobo. *Programación Extrema*. 2013.
20. Merín, Mauricio. *Curso PHP Advanced and Ajax*. 2012.
21. Rubira, Jorge. *Google Libraries API, un repositorio de las librerías Javascript más utilizadas*. 2011.
22. Ojeda, Antonio Navajas. *Guía completa de CSS3*.
23. Gauchat, Juan Diego. *El gran libro de HTML5, CSS3 y Javascript*.
24. *Evaluación de Frameworks para Aplicaciones Web* . Funes, Ana y Dasso, Aristides. San Luis, Argentina : s.n., 2014.
25. Eguiluz, Javier. *Desarrollo web ágil con Symfony2*. 2013.
26. Abreu, Pavel Javier Suárez. *Módulo estadístico para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel v1.0*. Habana : s.n., 2013.
27. León, Eduardo. *Tutorial Visual Paradigm for UML*. 2010.
28. Manual de MySQL. [En línea] mysql.com/documentation/index.html .
29. Manual de PostgreSQL. [En línea] mysql.com/documentation/index.html .
30. Otero, A. *MySQL vs PostgreSQL. ¿Cuándo emplear cada una de ellas?* 2007.
31. Versión 2.2 de la documentación del Servidor de HTTP Apache. [En línea] 2011. [Citado el: 15 de enero de 2015.] <http://httpd.apache.org/docs/2.2/>.
32. Programación Desarrollo. [En línea] [Citado el: 20 de enero de 2015.] <http://programaciondesarrollo.es>.
33. Sitio Oficial de Netbeans. [En línea] https://netbeans.org/index_es.html.
34. BILLSUS , DANIEL y PAZZANI, MICHAEL J. *User Modeling for Adaptive News Access*. 2000.
35. Segaran, Toby. *Programming Collective Intelligence* . 2005.
36. Manning, C. *An introduction to information retrieval*. Cambridge: Cambridge University Press : s.n., 2008.
37. Patrón Modelo-Vista-Controlador. [En línea] 2012. [Citado el: 13 de Marzo de 2015.] <http://revistatelematica.cujae.edu.cu>. 1.
38. Pressman, Roger. *Ingeniería del Software, Un Enfoque Práctico*.
39. Abreu Jiménez, Pedro Carlos y Chang, Jeny . *Base de Casos para estimar la duración de un proyecto de desarrollo de software*. Habana : s.n., 2012.
40. Murphey, Rebecca. *Fundamentos de jQuery*. 2013.
41. Martínez, R. *Portal sobre PostgreSQL en español*. 2010.
42. Martínez Sánchez, Natalia y García Lorenzo, María Matilde. *Mapa Conceptual interactivo-individualizado de apoyo al proceso de enseñanza aprendizaje de la teoría básica del enfoque lógico combinatorio*. Habana,Cuba : s.n., 2012.

43. Moya Ricardo, Raubel. *LA FORMACIÓN DEL PROFESIONAL DESDE UNA CONCEPCIÓN PERSONALIZADA DEL PROCESO DE APRENDIZAJE*. Guantánamo, Cuba : s.n., 2011.

44. *Sistema Tutorial Inteligente para el desarrollo de habilidades en la solución de problemas matemáticos*. Correa, Yharlan Alex Rojas. Bogotá, Colombia : s.n., 2007.