

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



FACULTAD 2

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.

“Sistema Tutorial Inteligente para la asignatura
Reconocimiento de Patrones.”

Autores: Lourdes Alicia Risco Ferrer
Luis Ernesto Manchón Lauzardo

Tutores: Ing. Maidelis Milanés Luque
Ing. Víctor Alejandro Roque Domínguez

Consultante: DrC. Natalia Martínez Sánchez

La Habana, junio de 2017

“Año 59 de la Revolución”



“Se pueden adquirir conocimientos y conciencia a lo largo de toda la vida, pero jamás en ninguna otra época de su existencia una persona volverá a tener la pureza y el desinterés con que, siendo joven, se enfrenta a la vida.”

Fidel Castro Ruz

Declaración de Autoría

Se declara que Lourdes Alicia Risco Ferrer y Luis Ernesto Manchón Lauzardo son los únicos autores del presente trabajo de Diploma que tiene por título “Sistema Tutorial Inteligente para la asignatura Reconocimiento de Patrones” y se concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, para hacer uso en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2017

Firma del Autor.

Lourdes Alicia Risco Ferrer

Firma del Autor.

Luis Ernesto Manchón Lauzardo

Firma del Tutor.

Ing. Maidelis Milanés Luque

Firma del Tutor.

Ing. Víctor Alejandro Roque
Domínguez

Agradecimiento

Les agradecemos a nuestros padres, por todo el apoyo incondicional que nos han brindado en estos años que hemos cursado en la universidad. Les agradecemos además a mi abuela Gema, a nuestra tía Marlenys, a mi hermano Jaime, a nuestro tío George, a nuestra tía Lourdes, a nuestras primas Loana y Janine. A nuestros tutores Maidelis y Víctor y demás profesores que han contribuido a nuestra formación como profesionales y a todos nuestros amigos en estos 5 años de estudios.

Resumen

El constante desarrollo de las Tecnologías de la Informática y las Comunicaciones (TICs), permite la búsqueda de soluciones para acelerar el avance de la sociedad. Uno de los sectores que ha evolucionado de conjunto con las TICs son los sistemas de educación, utilizando diferentes herramientas educativas en el proceso de enseñanza aprendizaje, desde la educación primaria hasta la superior. La Universidad de las Ciencias Informáticas (UCI) presenta una infraestructura adecuada para el desarrollo de herramientas tecnológicas que apoyen el proceso de enseñanza-aprendizaje.

El objetivo de este trabajo, comprende el desarrollo de un Sistema Tutorial Inteligente (STI) para la asignatura optativa Reconocimiento de Patrones (RP). El sistema se diseñó según la arquitectura de la Doctora Zulma Cataldi, por lo que quedó estructurado de la siguiente manera: Módulo del Dominio, Módulo Tutor, Módulo del Estudiante (está investigación recibe ya el estudiante clasificado, por lo que este módulo, solo realiza el diagnóstico cognoscitivo del estudiante) y la Interfaz de usuario. Se utilizó para su desarrollo como técnica de Inteligencia Artificial (IA) los Sistemas Basados en Reglas.

Palabras Claves: Herramientas Educativas, Reconocimiento de Patrones, Sistemas Basados en Reglas, Sistema Tutorial Inteligente.

Índice

Introducción.....	1
Capítulo I. Fundamentación Teórica	5
Introducción.....	5
1.1. Herramientas educativas	5
1.2. Sistemas Tutoriales Inteligentes	6
1.3. Sistemas Tutoriales Inteligentes en la educación.....	9
1.4. Técnicas de Inteligencia Artificial.....	14
1.5. Metodologías de Desarrollo de Software.....	15
1.5.1. Selección de la metodología.....	17
1.5.2. Lenguaje de Modelado	19
1.6. Plataformas de desarrollo.....	19
1.6.1. Lenguajes de programación	19
1.7. Herramientas de Desarrollo.....	20
1.7.1. Entorno de Desarrollo Integrado.....	20
1.7.2. Sistema Gestor de Base de Datos.....	21
1.7.3. Herramientas CASE.....	21
1.8. Servidor web	22
Conclusiones Parciales:.....	23
Capítulo II. Propuesta de solución. Exploración y Planificación.	24
Introducción.....	24
2.1. Propuesta de solución	24
2.2. Diseño del Sistema Basado en Reglas.....	27
2.3. Usuarios relacionados con el sistema.....	30
2.4. Fase de exploración	31
2.4.1. Historias de Usuario	31
2.4.2. Requisitos no funcionales del sistema	32
2.5. Fase de planificación.....	33
2.5.1. Plan de iteraciones	33
Conclusiones Parciales:	34
Capítulo III. Diseño, implementación y Prueba.....	35
Introducción.....	35

3.1. Fase de diseño	35
3.1.1. Tarjetas CRC.....	35
3.1.2. Estilo Arquitectónico	37
3.1.3. Patrones de diseño	40
3.2. Modelo Físico de la Base de Datos	41
3.3. Despliegue del sistema	42
3.4. Tareas de Ingeniería.....	42
3.5. Pruebas	45
3.6. Resultado de las pruebas realizadas.....	46
Conclusiones Parciales:.....	48
Conclusiones.....	49
Recomendaciones.....	50
Referencias Bibliográficas	51
Bibliografía	55
Anexos	59
Anexo 1. Historias de Usuario.....	59
Anexo 2. Tareas de Ingeniería.....	64
Anexo 3. Ejemplos de Casos de pruebas de aceptación.....	77

Índice de Tablas

Tabla 2. Usuarios relacionados con el sistema.....	30
Tabla 3. HU: Insertar protocolo pedagógico	32
Tabla 4. Plan de Iteraciones	33
Tabla 6. Tarjeta CRC: Gestionar Estudiante.....	36
Tabla 7. Tarjeta CRC: Gestionar Bibliografía.....	36
Tabla 8. Tarjeta CRC: Gestionar Cuestionario.....	36
Tabla 9. Tarjeta CRC: Gestionar Protocolo.....	37
Tabla 10. Tarea de Ingeniería 13. Listar la bibliografía de la base de datos.....	45
Tabla 11. Caso de prueba de aceptación HU14_P1. Insertar Protocolo.....	46
Tabla 12. Resultado de las pruebas	47
Tabla 13. HU: Autenticar Usuario.	59
Tabla 14. HU: Insertar Usuario.	59
Tabla 15. HU: Modificar Usuario.....	59
Tabla 16. HU: Listar Usuario.....	59
Tabla 17. HU: Eliminar Usuario.....	60
Tabla 18. HU: Insertar Estudiante.....	60
Tabla 19. HU: Listar Estudiantes.	60
Tabla 20. HU: Buscar Datos del Estudiante.....	60
Tabla 21. HU: Insertar Bibliografía.....	61
Tabla 22. HU: Mostrar Bibliografía.....	61
Tabla 23. HU: Eliminar Bibliografía.	61
Tabla 24. HU: Insertar Cuestionario.....	61
Tabla 25. HU: Mostrar Cuestionario.....	62

Tabla 26. HU: Eliminar Cuestionario.....	62
Tabla 27. HU: Mostrar Protocolo Pedagógico.....	62
Tabla 28. HU: Eliminar Protocolo Pedagógico.....	62
Tabla 29. HU: Modificar Protocolo Pedagógico.....	63
Tabla 30. HU: Asignar Protocolo Pedagógico.....	63
Tabla 31. Tarea de Ingeniería 1. Comprueba que el usuario y la contraseña existan en la base de datos.....	64
Tabla 32. Tarea de Ingeniería 2. Le asigna el rol según los permisos que tenga el usuario.....	64
Tabla 33. Tarea de Ingeniería 3. Comprobar que el usuario no esté creado en la base de datos.....	65
Tabla 34. Tarea de Ingeniería 4. Guardar los datos en la base de datos.....	65
Tabla 35. Tarea de Ingeniería 5. Buscar el usuario en la base de datos.....	65
Tabla 36. Tarea de Ingeniería 6. Modificar el rol que ocupa el usuario en la base de datos.....	66
Tabla 37. Tarea de Ingeniería 7. Listar todos los usuarios del sistema.....	66
Tabla 38. Tarea de Ingeniería 8. Buscar el usuario de la base de datos.....	67
Tabla 39. Tarea de Ingeniería 9. Eliminar el usuario de la base de datos.....	67
Tabla 40. Tarea de Ingeniería 10. Comprobar que el estudiante no exista en la base de datos.....	67
Tabla 41. Tarea de Ingeniería 11. Guardar datos en la base de datos y mostrar sección correspondiente.....	68
Tabla 42. Tarea de Ingeniería 12. Salir del sistema en cualquier momento.....	68
Tabla 43. Tarea de Ingeniería 13. Mostrar todos los datos de todos los estudiantes en la base de datos.....	69
Tabla 44. Tarea de Ingeniería 14. Mostrar todos los estudiantes en la base de datos que cumplen con el criterio de búsqueda.....	69

Tabla 45. Tarea de Ingeniería 15. Seleccionar la bibliografía para subir al sistema.....	69
Tabla 46. Tarea de Ingeniería 15. Rellenar los campos de clasificación de la bibliografía.....	70
Tabla 47. Tarea de Ingeniería 17. Listar bibliografía de la base de datos.	70
Tabla 48. Tarea de Ingeniería 18. Muestra toda la bibliografía existente en el sistema.....	71
Tabla 49. Tarea de Ingeniería 20. Buscar que el cuestionario no esté creado en la base de datos.	71
Tabla 50. Tarea de Ingeniería 21. Crear el nuevo cuestionario en la base de datos.	71
Tabla 51. Tarea de Ingeniería 22. Listar todos los cuestionarios existentes en la base de datos.....	72
Tabla 52. Tarea de Ingeniería 23. Eliminar el cuestionario específico existente en la base de datos.	72
Tabla 53. Tarea de Ingeniería 24. Listar bibliografía de la base de datos.	73
Tabla 54. Tarea de Ingeniería 25. Insertar la bibliografía necesaria para ese protocolo pedagógico.	73
Tabla 55. Tarea de Ingeniería 26. Buscar que el protocolo pedagógico no este creado.	73
Tabla 56. Tarea de Ingeniería 27. Listar todos los protocolos pedagógicos existentes en la base de datos.	74
Tabla 57. Tarea de Ingeniería 28. Eliminar protocolo pedagógico específico existente en la base de datos.....	74
Tabla 58. Tarea de Ingeniería 29. Comprobar que el protocolo pedagógico esté creado en la base de datos.	75
Tabla 59. Tarea de Ingeniería 30. Modificar los datos en la base de datos.....	75

Tabla 60. Tarea de Ingeniería 31. Asigna el protocolo pedagógico correspondiente al estudiante.	76
Tabla 75. HU16_P1. Mostrar Protocolo Pedagógico	77
Tabla 76. HU17_P1. Eliminar Protocolo Pedagógico.....	77
Tabla 77. HU18_P1. Modificar Protocolo Pedagógico.	78
Tabla 78. HU19_P1. Asignar Protocolo Pedagógico.	78

Índice de Figuras

Figura 1. Interacción entre los módulos de un STI. Fuente: Fernando Lage y Zulma Cataldi. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia.	7
Figura 2. Módulo del Dominio. Fuente: Fernando Lage y Zulma Cataldi. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia.	7
Figura 3. Módulo de Tutor. Fuente: Fernando Lage y Zulma Cataldi. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia.	8
Figura 4. Módulo del Estudiante	9
Figura 5. Técnicas de IA utilizadas en los STI estudiados	13
Figura 6. Método de la Estrella de Bohem-Turner.....	17
Figura 7. Propuesta de solución del Sistema. Fuente: Elaboración propia.	24
Figura 8. Propuesta de solución del Sistema. Fuente: Elaboración propia.	28
Figura 9. Vista del Modelo	38
Figura 10. Vista de Vista.....	39
Figura 11. Vista del Controlador	40
Figura 12. Modelo físico de la Base de Datos	41
Figura 13. Diagrama de Despliegue	42
Figura 14. Pruebas de Aceptación	48

Introducción

Con el desarrollo de las Tecnologías de la Informática y las Comunicaciones (TICs) el énfasis de la profesión docente está cambiando, desde un enfoque centrado en el profesor que se basa en prácticas alrededor de la pizarra y las conferencias, basado en clases magistrales, hacia una formación centrada principalmente en el alumno dentro de un entorno interactivo de aprendizaje. El diseño e implementación de programas de capacitación docente que utilicen las TICs, son un elemento clave para lograr reformas educativas profundas y de amplio alcance. [1]

En las universidades y centros de enseñanza es palpable el incremento de la utilización de las TICs en los procesos de enseñanza-aprendizaje. En este contexto, las más demandadas son las tecnologías basadas en Inteligencia Artificial (IA), como los tutores inteligentes, los sistemas de gestión del aprendizaje y otros. [2]

La Universidad de las Ciencias Informáticas (UCI), presenta una infraestructura adecuada para el desarrollo de herramientas tecnológicas que apoyen el proceso de enseñanza-aprendizaje. “ZERA”, es el entorno virtual de aprendizaje que sustenta las asignaturas que componen el plan de estudio de la Universidad, incluyendo Reconocimiento de Patrones (RP), asignatura optativa que se oferta en 4to y 5to año.

ZERA es una plataforma educativa innovadora, interactiva y adaptable, que reúne características de Sistemas de Gestión de Contenidos de Aprendizaje, apoyando el proceso de enseñanza-aprendizaje en instituciones educativas, a partir de un conjunto de herramientas centradas en los aprendices y ambientes de aprendizaje colaborativo [3]. La plataforma permite a los profesores la gestión por asignaturas de la bibliografía, ejercicios evaluativos, la creación de foros debates, chats, trabajos independientes, entre otras actividades. Una de las potencialidades de la herramienta es la posibilidad que brinda de poder crear espacios privados para una facultad en particular.

A pesar de todas las posibilidades que brinda la plataforma establecida en la Universidad, los estudiantes que cursan la asignatura Reconocimiento de

Patrones presentan una serie de dificultades a la hora de la realización del estudio independiente que a continuación se detallan: desconocen según sus características individuales, cuál sería la bibliografía más acertada para la apropiación del conocimiento, presentan deficiencias para autoevaluarse en una temática supuestamente ya vencida. Además necesitan la presencia constante del profesor para que les oriente el camino a seguir y los mantenga al tanto de su situación docente actual.

Estas dificultades no pueden ser resueltas con la plataforma en cuestión, dado que la misma no permite de manera automática, la creación de estrategias de aprendizaje atendiendo a las características individuales de cada estudiante. Además tampoco permite la retroalimentación y la actualización del estado de conocimiento del alumno sin la intervención del profesor.

Partiendo de la situación problemática antes mencionada se definió como **problema a resolver**: La herramienta educativa existente en la Universidad no tiene en cuenta las características individuales de los estudiantes para guiar el proceso de enseñanza aprendizaje.

En consecuencia con lo anterior, se determinó como **objeto de la investigación**: las Herramientas Educativas y como **campo de acción**: los Sistemas Tutoriales Inteligentes.

Por tanto, el **objetivo de la investigación** es: desarrollar un Sistema Tutorial Inteligente para la asignatura Reconocimiento de Patrones en la Facultad 2 de la Universidad de las Ciencias Informáticas.

Objetivos Específicos:

- Explicar los referentes teóricos sobre los Sistemas Tutoriales Inteligentes como herramienta educativa.
- Caracterizar las herramientas, lenguajes y metodología para la propuesta de solución.
- Diseñar las Historias de Usuario, los Requisitos no Funcionales del Sistema y las Tarjetas CRC del Sistema Tutorial para la asignatura de Reconocimiento de Patrones en la Universidad de las Ciencias Informáticas.

- Implementar el Sistema Tutorial Inteligente según la metodología XP.
- Aplicar las pruebas para verificar el funcionamiento del Sistema Tutorial Inteligente propuesto.

Para darle solución a los objetivos planteados se proponen las siguientes **tareas de la investigación:**

- Exponer los referentes teóricos de los Sistemas Tutoriales Inteligentes como herramienta educativa para establecer la posición del investigador.
- Analizar las diferentes técnicas de Inteligencia Artificial aplicadas en los módulos de los Sistemas Tutoriales Inteligentes y sus características, para seleccionar la más adecuada según la problemática de la investigación.
- Analizar las herramientas, lenguajes y metodología para la propuesta de solución.
- Seleccionar las herramientas, lenguajes y metodología acordes para la propuesta de solución del Sistema Tutorial Inteligente.
- Explicar las fases de Diseño, Implementación y Prueba de un Sistema Tutorial Inteligente, para la asignatura Reconocimiento de Patrones en la Facultad 2 de la Universidad de las Ciencias Informáticas.
- Aplicar las pruebas al Sistema Tutorial Inteligente para verificar su correcto funcionamiento.

Los **métodos teóricos** utilizados en la investigación son:

Análisis-Síntesis: permitió la obtención de información teórica acerca de los diferentes conceptos y aspectos relacionados con los STI, los módulos que lo conforman y su utilización en la educación como herramienta educativa. También facilitó el estudio de las metodologías, lenguajes y herramientas adecuadas para el desarrollo del Sistema Tutorial propuesto.

Modelación: se utilizó para mostrar mediante un diagrama la propuesta de solución, así como los diagramas de las etapas de exploración, planificación, diseño, implementación y prueba según la metodología utilizada.

Del nivel **empírico** se utilizaron los siguientes:

Documental: se utilizó para consultar bibliografía en fuentes de carácter documental, tales como libros y artículos.

Entrevista: fueron realizadas a los profesores de la asignatura Reconocimiento de Patrones, para conocer las necesidades del sistema.

Para lograr una mejor organización y lectura del trabajo de diploma, este se estructuró de la siguiente manera: Resumen, Introducción, Tres Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía y Glosario de términos, donde se abarca todo lo relacionado con la investigación realizada.

- **Capítulo 1. Fundamentación Teórica:** se realiza un estudio de las diferentes formas en que los Sistemas Tutoriales Inteligentes son utilizados como herramientas educativas, abordándose los conceptos y aspectos más importantes para la investigación. Además, se realiza un análisis sobre la metodología utilizada, las técnicas más utilizadas en el desarrollo del sistema, así como las herramientas y tecnologías necesarias para su implementación.
- **Capítulo 2. Exploración y Planificación:** En el capítulo se exponen la propuesta de solución diseñada, las características funcionales y no funcionales del sistema, además de algunos de los artefactos correspondientes que requiere la metodología de desarrollo.
- **Capítulo 3. Diseño, Implementación y Prueba:** Se describen las fases de diseño, implementación y prueba, definidas por la metodología XP, donde se detallan las tareas de ingeniería generadas por cada HU, se describen los patrones de arquitectura utilizados y las tarjetas CRC. Además, se muestran los resultados de las pruebas aplicadas, pues al finalizar cada iteración, el producto debe ser probado para verificar si cumple con las necesidades exigidas por el cliente.

Capítulo I. Fundamentación Teórica

Introducción

En el presente capítulo se analizan las herramientas educativas, haciendo énfasis en los STI. Se exponen los resultados del análisis de la arquitectura de diez STI, así como las técnicas de IA que aplican. Se define la técnica de IA a utilizar para dar solución al problema planteado. Por último, se analiza la metodología, lenguajes y herramientas seleccionadas para el desarrollo del sistema.

1.1. Herramientas educativas

Existen diferentes terminologías para denominar los recursos tales como: recursos de aprendizajes, recursos didácticos, materiales didácticos, medios didácticos, herramientas educativas, aplicaciones digitales para la educación, entre otros.

La definición planteada por Rabajoli – Ibarra (2008 p.46), el cual plantea que un recurso puede ser un contenido que implica información y/o un software educativo, caracterizado éste último, no solamente como un recurso para la educación sino para ser utilizado de acuerdo a una determinada estrategia didáctica. [4]

Las herramientas educativas están diseñadas con el fin de apoyar la labor de los profesores en el proceso de enseñanza-aprendizaje, permitiendo el desarrollo de habilidades cognitivas en el estudiante. Entre sus principales características se encuentran:

- Fomento de la comunicación profesor/alumno: la relación profesor/alumno, al transcurso de la clase o a la eventualidad del uso de las tutorías, se amplía considerablemente con el empleo de las herramientas de la plataforma virtual. El profesor tiene un canal de comunicación con el alumno permanentemente abierto.
- Facilidades para el acceso a la información: permite crear y gestionar asignaturas de forma sencilla, incluir gran variedad de actividades y hacer un seguimiento exhaustivo del trabajo del alumnado. Además permite al alumno

remitir sus actividades y que estas queden almacenados en la base de datos.

- Fomento del debate y la discusión: propicia que el alumno pueda examinar una materia, conocer la opinión al respecto de otros compañeros y exponer su propia opinión, al tiempo que el profesor puede moderar dichos debates y orientarlos. [5]

Dentro de los grupos de clasificaciones de las herramientas educativas se encuentran las que utilizan inteligencia artificial y las que no, dentro de las que utilizan IA se encuentran los SGA y los STI.

1.2. Sistemas Tutoriales Inteligentes

Los Sistemas Tutores Inteligentes (STI) se definen como software que aplican técnicas de Inteligencia Artificial para poder proporcionar de forma gráfica el conocimiento a los alumnos. Se dice que son inteligentes gracias al hecho de poder adaptarse a las necesidades de cada estudiante. [6]

Los STI deben soportar lo que se denomina procesamiento distribuido, para poder utilizar la red de una manera eficiente, ya sea la Internet o cualquier otra red. La provisión de información útil y el mejoramiento de la experiencia educativa a través de nuevos recursos pedagógicos facilita el proceso cognitivo del estudiante. A esto se le puede sumar, un entorno flexible con acceso a la Internet desde cualquier punto en que el STI se encuentre almacenado. Esto significa no solo una reducción de costos importantes (materiales y tiempo principalmente) sino una mejora sustancial en el uso de las plataformas para la educación.

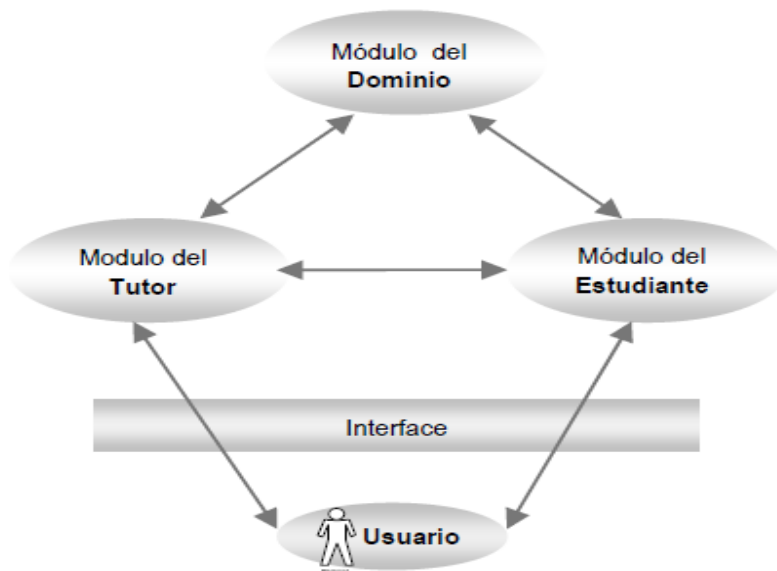


Figura 1. Interacción entre los módulos de un STI. Fuente: Fernando Lage y Zulma Cataldi. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia.

Los STI permiten la emulación de un tutor humano a fin de poder determinar: qué enseñar, cómo enseñar y a quién enseñar a través del **Módulo Dominio** el cual tiene el objetivo global de almacenar todos los conocimientos dependientes e independientes del campo de aplicación del STI.

Básicamente deberá tener los sub-módulos siguientes:

- Conocimientos: son los contenidos que deben cargarse en el sistema, a través de los conceptos, las preguntas, los ejercicios, los problemas y las relaciones.
- Elementos Didácticos: son las imágenes, videos, sonidos, es decir, material multimedia que se requiere para facilitarle al alumno apropiarse de conocimiento en la sesión pedagógica. [6]

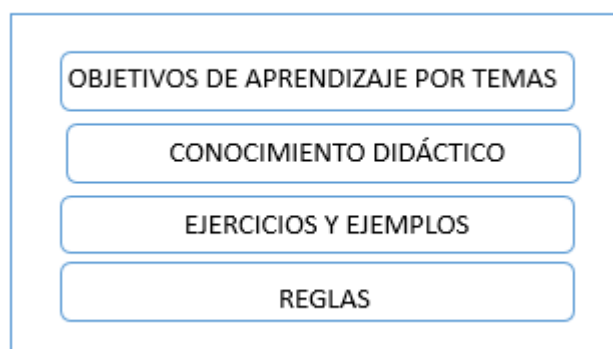


Figura 2. Módulo del Dominio. Fuente: Fernando Lage y Zulma Cataldi. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia.

Módulo del Tutor:

Define y aplica una estrategia pedagógica de enseñanza. Contiene los objetivos a ser alcanzados. Selecciona el material de aprendizaje para el estudiante.

Consta de:

Protocolos Pedagógicos, según la Dra. Zulma Cataldi [7] son almacenados en la base de datos y se clasifican en:

- Socrático: este método espera que sea el mismo estudiante el que realice los descubrimientos por iniciativa propia.
- Magistral: se trata de una exposición continua de un conferenciante. Los alumnos solo escuchan y toman notas. El proceso de comunicación es únicamente entre un profesor que desarrolla un papel activo y los alumnos que son receptores pasivos de la información.
- Entrenamiento: ofrece una respuesta, ya que hace hincapié en la práctica reflexiva y la realimentación informativa. Las principales actividades del docente entrenador consisten en asignar prácticas, alentar a los alumnos a reflexionar sobre lo que están haciendo y ofrecer la realimentación.

Planificador de lecciones: organiza los contenidos de la misma.

Analizador de Perfil: analiza las características del alumno, seleccionando la estrategia pedagógica más conveniente.

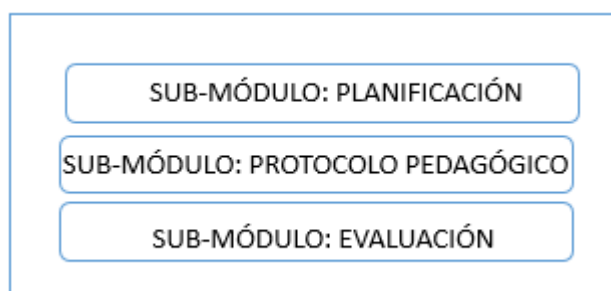


Figura 3. Módulo de Tutor. Fuente: Fernando Lage y Zulma Cataldi. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia.

Módulo del Estudiante

Tiene por objetivo el diagnóstico cognitivo del estudiante y el modelado del mismo. Está compuesto por los sub-módulos:

- Estilos de aprendizaje: compuesto por una base de datos con los estilos de aprendizajes disponibles en el sistema, los métodos de selección de estilos y las características de cada uno de ellos.

- Estado de Conocimiento: contiene el mapa de conocimientos obtenido inicialmente a partir del módulo del dominio. [8]

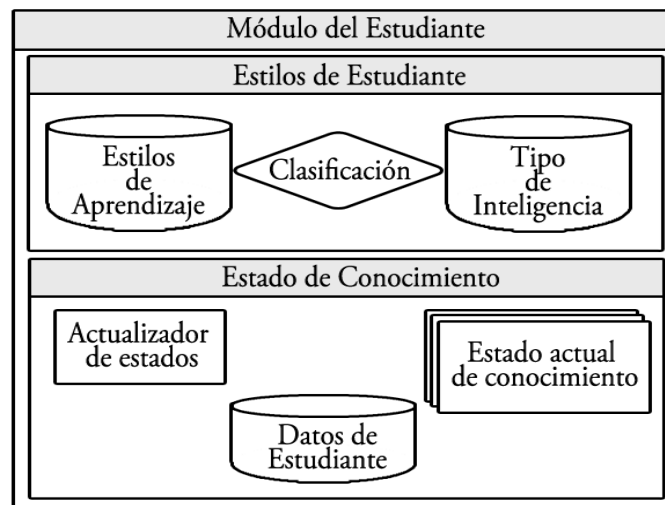


Figura 4. Módulo del Estudiante. Fuente: Fernando Lage y Zulma Cataldi. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia.

1.3. Sistemas Tutoriales Inteligentes en la educación

A continuación, se muestran un grupo de los sistemas tutoriales inteligentes existentes en el mundo, aplicados como herramientas educativas. Daremos una breve explicación para qué fueron desarrollados, cómo están estructurados y qué técnicas de Inteligencia Artificial utilizaron algunos de estos STI.

- Herramienta de Autoría **HEDEA**

Trabaja con un laboratorio virtual y permite desarrollar un STI a partir de la definición temática de un curso. Dichos STI se basan en modelos probabilísticos partiendo del temario pasado de un curso. La generación del modelo del estudiante se realiza de manera automática y es transparente al usuario. Los STI generados por la herramienta son asociados a un laboratorio virtual existente. El modelo del estudiante toma en cuenta los valores de experimentos previos lo cual permite darle mayor valor al historial o a su último resultado.

El sistema está conformado por cuatro módulos, módulo tutor, módulo estudiante y la interfaz; como técnica de Inteligencia Artificial utiliza Redes Bayesianas y Sistemas Basados en Regla. [9]

- **APA-Prolog**

Es un Ambiente de enseñanza-aprendizaje para la programación lógica que se adapta a los conocimientos previos del estudiante, elaborado por el grupo de

desarrollo de Software para la Educación de la Universidad de Sancti Spíritus “José Martí Pérez”.

Su interfaz la constituye un conjunto de mapas conceptuales relacionados entre sí y utiliza Razonamiento Basado en Casos (RBC) como técnica de Inteligencia Artificial.

- **HESEI**

Es una herramienta para elaborar Sistemas de Enseñanza-Aprendizaje Inteligentes que se adapten a las características de los estudiantes a través del uso de Mapas Conceptuales y Razonamiento Basado en Casos como técnicas de IA, elaborada en la Universidad Central “Marta Abreu” de Las Villas. [10]

- **STI-SO**

Desarrollado en la Universidad de las Ciencias Informáticas, es una herramienta de apoyo al proceso de enseñanza-aprendizaje de la asignatura Sistema Operativo, la cual ayuda a consolidar los contenidos del plan de estudio definido para la asignatura.

El STI-SO fue estructurado en cuatro módulos: Módulo del Estudiante, Módulo del Dominio, Módulo Pedagógico y la Interfaz. Este STI utiliza el RBC como técnica de IA para asignarle al estudiante el material de estudio. [11]

- **ELE-Tutor**

Es un Sistema Tutorial para el español como Lengua Extranjera, creado en la Universidad de Concepción en Chile, el diseño de cada módulo y los mecanismos como se articulan para conseguir el objetivo final del sistema, que es la adaptabilidad a las necesidades de cada estudiante en el contexto de la enseñanza del español como Lengua Extranjera.

Su arquitectura se compone, principalmente de tres partes: un módulo del experto, un módulo del tutor y un módulo del estudiante, cada uno de estos componentes forman parte de un sistema mayor (STI) que, al articularse, permiten la adaptabilidad del sistema al conocimiento del estudiante. Para lograr una articulación completa del sistema, es necesario junto con diseñar el modelo de cada parte, establecer los parámetros, variables e información que comparten cada uno ellos y la forma cómo llevan a cabo su tarea. Este STI utiliza como técnica de Inteligencia Artificial PLN denominado Generación de Lenguaje Natural se entiende como la construcción o producción de textos a través de sistemas inteligentes con el objeto de responder a propósitos comunicativos. [12]

- **Tutor Inteligente con reconocimiento y manejo de emociones para Matemáticas**

Es un Sistema Tutor Inteligente para matemáticas de tercer grado de primaria que identifica el estado emocional del estudiante y produce retroalimentación afectiva para el mismo durante un curso, el cual se encuentra instalado en una red social. La red social y el Sistema Tutor Inteligente con manejo afectivo han sido probados en escuelas públicas y privadas de la localidad, con resultados muy favorables.

El reconocimiento de emociones se lleva a cabo a través de expresiones faciales, lo cual se realiza por medio de una red neuronal artificial. El sistema tutor inteligente y afectivo para la red social adopta el modelo tradicional conocido como arquitectura de cuatro-módulos, donde el primer módulo (la interfaz de la red social) tiene acceso a otros tres módulos principales llamados: dominio, estudiante y tutor. [13]

- **Sistema Tutorial Inteligente para el Apoyo de la Enseñanza de la Lectura Inicial**

Se diseñó y desarrolló un software educativo para apoyar el proceso de aprendizaje de la lectura inicial, basado en los conocimientos actuales sobre enseñanza de la lectoescritura y en el modelo de Sistemas Instruccionales Inteligentes. El Software desarrollado se evaluó en una muestra de 8 niños de una escuela municipal de Santiago de Chile, identificados por la profesora como presentando algún grado de retraso lector. Se evaluaron los siguientes aspectos: claridad de las instrucciones, utilidad de los refuerzos y ayudas, sensibilidad al nivel de desempeño del niño y potencial como herramienta instruccional.

Para el desarrollo del software se escogió el modelo propuesto por Wengner (1987) de Sistema Instruccionales Inteligentes (ITS: Intelligent Tutoring System.), Modelo del dominio, Modelo del Alumno y Modelo Pedagógico. Se utilizó como técnica de Inteligencia Artificial un Sistema Basados en Reglas para regular el flujo de actividades asociadas a cada niño. [14]

- **Sistema de enseñanza-aprendizaje inteligente de apoyo al curso introductorio de Matemática (SEAI)**

Se desarrolló un STI de apoyo al proceso de enseñanza-aprendizaje de la matemática en el curso introductorio que se imparte en la Universidad de las Ciencias Informáticas. El desarrollo de un modelo basado en herramientas

inteligentes, posibilita la ayuda en la toma de decisiones en el diagnóstico de estudiantes en el proceso de enseñanza-aprendizaje de matemática. La población estuvo formada por trescientos cuatro (304) estudiantes, de los cuales se tomó una muestra del 60% que proporciona un tamaño muestral de ciento ochenta y dos (182) estudiantes.

El software se desarrolló según la arquitectura descrita en Ovalle, 2007, la cual reúne los elementos más comúnmente encontrados en la literatura consultada y se resumen en el criterio que plantea que un SEAI está compuesto por un módulo del dominio, un módulo del alumno y el módulo pedagógico, que operan de forma interactiva y se comunican a través de un módulo central que suele denominarse módulo entorno. Se utilizó un Sistema Basado en Conocimiento como técnica de Inteligencia Artificial, para el desarrollo de este software. [15]

- **Sistema Inteligente para el Álgebra Lineal (SIAL)**

El SIAL se creó con el objetivo de apoyar al estudiante durante su estudio independiente para brindarle la vía de solución de los problemas que debe resolver.

El sistema fue desarrollado guiándose por los fundamentos teóricos que están relacionados con las técnicas de Inteligencia Artificial para clasificar, utilizando el clasificador k-vecinos más cercanos K-NN (Caballero, 2010), así como los cinco problemas tipos del Álgebra Lineal (Yordi, 2004). Posee dos módulos llamados: “Obtener vía de solución para el tema de Espacios Vectoriales del Álgebra Lineal” y “Obtener vía de solución del sistema”, usando la base de conocimiento diseñada y el clasificador para inferir soluciones. La Técnica de Inteligencia Artificial utilizada para el desarrollo del software fue Razonamiento Basado en Casos. [16]

- **Sistema Tutor Inteligente Adaptativo para Laboratorios Virtuales y Remotos**

El sistema propone un nuevo enfoque para el diseño de un STI adaptativo aplicado a los estudios de Ingeniería de Sistemas y Automática a las prácticas de laboratorios virtuales y remotos. El modelo del sistema se obtiene de la aplicación de la teoría básica de control (realimentación) en combinación con técnicas de minería de datos que extraen y analizan la información de las interacciones del alumno con la plataforma o LMS en su proceso de enseñanza y aprendizaje.

Está compuesto por tres módulos: el modelo del alumno (evalúa el conocimiento y la actualización del alumno en el sistema), el modelo del tutor (selecciona la intervención educativa más adecuada al alumno) y el modelo dominio (contiene los recursos educativos del curso). Se utiliza la combinación de técnicas de minería de datos que mejor se adapta para el análisis y funcionamiento del sistema. [17]

En este epígrafe se realizó una breve selección de algunos STI utilizados para los diferentes niveles de enseñanza que existen en la educación y la gran utilidad de estos sistemas. De forma general podemos afirmar que la mayoría de los STI están compuestos por cuatro módulos: Módulo Tutor, Módulo Estudiante, Módulo Dominio e Interfaz. Las técnicas de IA utilizadas por los sistemas analizados en el estado del arte son: los Sistemas Basados en Casos, en Reglas, las Redes Neuronales, entre otras. A continuación, se presenta una gráfica donde se muestra qué técnica de IA fue la más utilizada entre los 10 STI investigados.

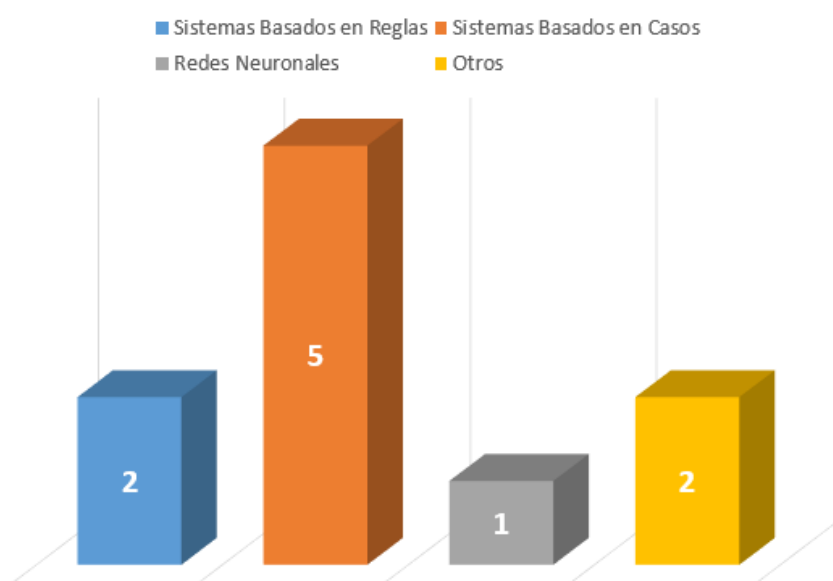


Figura 5. Técnicas de IA utilizadas en los STI estudiados

Como se observa en la Figura 5, el 50% de los STI estudiados utilizan los sistemas basados en casos como técnica de Inteligencia Artificial, no se pudo utilizar esta técnica porque no se cuenta con una base de datos histórica. Pero cuenta con la experiencia de los expertos, que permiten modelar su

conocimiento a partir de reglas, para seleccionar los protocolos pedagógicos a utilizar en los estudiantes.

1.4. Técnicas de Inteligencia Artificial

En los STI es necesario la aplicación de las técnicas de IA, tanto en la clasificación de los estudiantes, como en la asignación de bibliografías. En los distintos STI que se han investigado se puede observar que uno de los principales objetivos de aplicar la IA, es que el sistema logre asignar de manera autónoma la bibliografía y logre aprender sobre cómo debe asignar a los usuarios. A continuación, se describe en qué consiste la técnica de IA escogida para el desarrollo del sistema.

Sistemas Basados en Reglas

Un sistema basado en reglas (SBR) es un sistema basado 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 o premisa.

ENTONCES (THEN), es el consecuente o conclusión.

En este tipo de sistema, la BC contiene las variables y el conjunto de reglas que definen el problema, y el Motor de Inferencia (MI) obtiene las conclusiones aplicando la lógica a estas reglas. La Base de Casos contiene la representación del conocimiento sobre el dominio de aplicación del sistema y se divide en la base de hechos y en la base de reglas. 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 sea aplicable y en el consecuente el conjunto de acciones derivadas de la aplicación de la regla. El MI utiliza los datos y el conocimiento para obtener conclusiones o hechos. 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 reglas y 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, el encadenamiento de reglas y el encadenamiento de reglas orientado a un 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.

Los SBR emplean el forward Chaining: en el que los nuevos hechos son inferidos de hechos existentes y el backward Chaining: por ejemplo, si hay una regla IF A THEN B, aplicar backward chaining significa usar el conjunto de reglas existentes para determinar que debe ser cierto para que B también lo sea. Esta técnica es empleada para probar una hipótesis. [18]

1.5. Metodologías de Desarrollo de Software

Para la selección de la metodología de desarrollo a utilizar, se utilizó el método de la estrella de Boehm-Turner para decidir qué tipo de metodología era la más acorde para el desarrollo del sistema.

El método de la estrella de Boehm-Turner, es un modelo de fácil comprensión capaz de evaluar, cuantificar e identificar 5 variables críticas a la hora de decidir si el desarrollo de un sistema se aventura por metodologías ágiles o robustas. Estas variables críticas son las siguientes:

- Personal:

El por ciento del personal con capacidad, experiencia y posibilidades para enfrentar tareas es un aspecto de mucha importancia en un proyecto de producción de software, de esto depende mucho la posibilidad que tiene el grupo de desarrolladores de poder implantar o desarrollar un tipo de metodología u otra.

En esta variable el equipo de desarrollo está clasificado en un 20% Junior, según la variable Personal, dado que posee un nivel de experiencia en el desarrollo de aplicaciones web.

- Criticidad:

La criticidad del sistema resultada determinante cuando se habla de proyectos cuya falla puede traer pérdidas de vidas humanas, daño al medio ambiente y grandes pérdidas económicas, por lo que se hace necesario emplear métodos robustos capaces de mantener un rigor de requisitos y diseño adecuados para procesos de pruebas, verificación y validación.

En esta variable se clasificó el sistema en utilidades, dado que, ante cualquier falla, el sistema no representa peligro para las vidas humanas o para el medio ambiente y no tiene repercusión económica.

- **Cultura:**

El por ciento del personal con capacidad para enfrentarse a entornos caóticos resulta importante a la hora de adoptar métodos ágiles los cuales se basan fundamentalmente en el talento de las personas, un ambiente laboral con un control excesivamente normalizado y jerarquizado resultaría incómodo para llevar a cabo este tipo de práctica.

El equipo de trabajo se clasificó con un 90%, según la variable, dado que el equipo de trabajo está compuesto por dos desarrolladores, con experiencia en el desarrollo de aplicaciones web y además de que el equipo de trabajo lleva conformado hace tres años.

- **Tamaño:**

La cantidad de personas de un proyecto resulta importante en dependencia de qué método ágil o robusto se vaya a poner en práctica. Para métodos ágiles donde es muy significativa una buena comunicación entre todos los integrantes del grupo de desarrollo esta resulta imposible con un número alto de personal en el proyecto.

El equipo de trabajo está conformado por 2 personas que se encargan de realizar todas las tareas para el desarrollo de la aplicación.

- **Dinamismo:**

El por ciento de cambios de requerimientos que pueden ocurrir en un mes tiene gran peso en los métodos ágiles preparados para enfrentar este tipo de situación, resultaría muy incómodo enfrentar grandes cambios de requerimientos con métodos robustos que tienen grandes volúmenes de documentación y una verificación y validación de procesos continua.

El equipo de trabajo se clasificó con un 50%, según la variable, debido a las necesidades del cliente que se han estado actualizando en cada reunión con el cliente. [19]

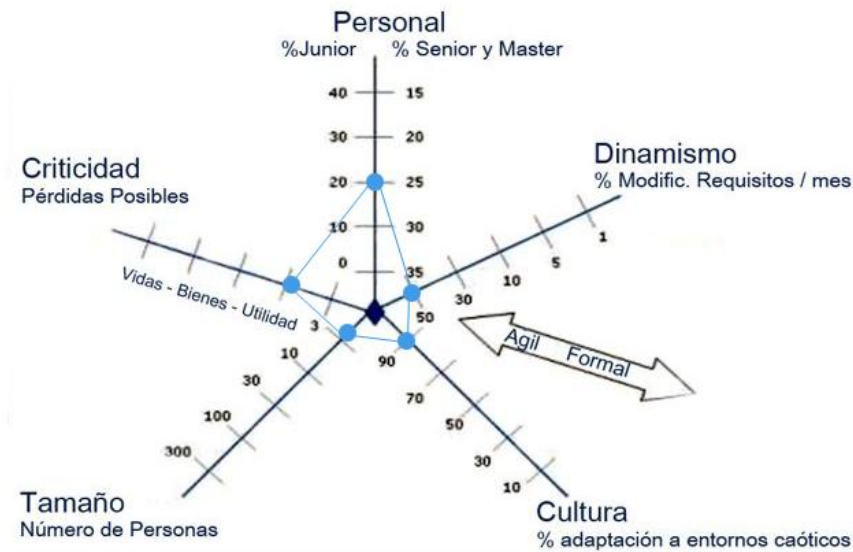


Figura 6. Método de la Estrella de Boehm-Turner

El método de la estrella de Boehm-Turner arrojó según las variables dinamismo, cultura, tamaño, criticidad y personal que el tipo de metodología más adecuada a aplicar es la ágil.

1.5.1. Selección de la metodología

Se seleccionó la metodología ágil, Programación Extrema (XP, por sus siglas en inglés) para la implementación del sistema, porque el equipo de desarrollo ya tiene experiencia trabajando con esta metodología en años anteriores.

XP es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Es utilizada en proyectos con requisitos muy cambiantes ya que se basa en proyectos pequeños, caracterizados por su sencillez, tanto en el aprendizaje como en su aplicación. Es una metodología ligera, flexible, predecible, de bajo riesgo y no por ello menos científica. Entre otras ventajas pueden mencionarse los pocos requerimientos de documentación y planificación, así como la exigencia de tener siempre el cliente disponible para el desarrollo, implicando una mejor correspondencia entre el producto y la necesidad del negocio, promoviendo y propiciando un buen clima de trabajo en equipo. Uno de los objetivos fundamentales de XP es la satisfacción del cliente, ya que la metodología trata de brindar el software que este necesite y cuando lo necesite.

Por tanto, se debe responder muy rápido a sus necesidades, incluso cuando los cambios sean al final de ciclo de la programación.

Fases de la metodología:

Exploración: el equipo de desarrollo cuenta con suficiente material de trabajo, se realizan las historias de usuarios (HU), que se recopilan en esta etapa, como para producir una primera entrega. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y la destreza que tengan los programadores con la tecnología.

Planificación: es una fase corta, en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario, las cuales son organizadas en iteraciones. También se define la fecha de las entregas de las historias de usuarios, las cuales deben entregarse en menos de tres meses.

Diseño: hace especial énfasis en los diseños simples y claros. Los conceptos de diseño en esta metodología utilizados para el desarrollo del sistema son los siguientes:

Simplicidad: un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

Recodificación: la recodificación (“refactoring”) consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible.

Pruebas: se dividen en dos grupos: pruebas unitarias, desarrolladas por los programadores, con el objetivo de verificar el código y las pruebas de aceptación para evaluar si al final de cada iteración se obtuvo la funcionalidad requerida, comprobando que sea la esperada por el cliente. [20]

1.5.2. Lenguaje de Modelado

El Lenguaje Unificado de Modelado (Unified Modeling Language UML), es un lenguaje estándar para escribir planos de software, UML se puede utilizar para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. UML prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.[21]

1.6. Plataformas de desarrollo

Una de las características fundamentales de las aplicaciones web, es el uso de la arquitectura Cliente–Servidor, la cual, describe con claridad la filosofía de funcionamiento de este tipo de aplicaciones. A continuación, se describen las tecnologías seleccionadas para la implementación del sistema.

1.6.1. Lenguajes de programación

PHP v5.5.12

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. [22]

Características:

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL

- Posee una amplia documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.

JavaScript v1.4.2

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. [23]

1.7. Herramientas de Desarrollo

A continuación, se detallan todas las herramientas de desarrollo que fueron necesarias utilizar para la planificación e implementación del STI, como el entorno de desarrollo integrado, el gestor de base de datos y la herramienta CASE.

1.7.1. Entorno de Desarrollo Integrado

PhpStorm v7.1.3

Es un Entorno de Desarrollo Integrado (IDE) que proporciona un editor para PHP, HTML y JavaScript con análisis de código al instante, prevención de errores y refactorizaciones automatizadas para PHP y código JavaScript. La terminación de código de PhpStorm soporta desde la versión de PHP 5.3, hasta la 7.0, incluyendo subrutinas y palabras claves. Incluye un editor SQL completo con resultados de consulta editables. Los usuarios pueden extender el IDE mediante la instalación de complementos creados para la plataforma IntelliJ o escribir sus propios complementos.

Características:

- Editor de código inteligente PHP.
- Calidad de análisis en el código.

- Entorno de desarrollo.
- Editor HTML/CSS/JavaScript.
- Depuración y pruebas.
- Experiencia multiplataforma.

1.7.2. Sistema Gestor de Base de Datos

MySQL v5.7.14

Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. [24]

Entre las características disponibles en las últimas versiones se pueden destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica y transacciones.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto (Oracle, 2016).

1.7.3. Herramientas CASE

En *Terminology for Software Engineering and Computer-aided Software Engineering* by B.Terry & D.Logee, *Software Engineering Notes*, Abril 1990, CASE es definido como:

“Herramientas individuales para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software (o mantenimiento). [25]

Visual Paradigm v8.0

Es una herramienta CASE de código abierto para el desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. [26]

Características principales:

- Entorno de creación de diagramas para UML
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad en múltiples plataformas.

La Herramienta Visual Paradigm se utilizó para el diseño de todos los procesos ingenieriles que fueron necesarios para el desarrollo del STI. También se utilizó para el diseño de la base datos.

1.8. Servidor web

Es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (text, widgets, banners, etc.). Estos servidores web utilizan el Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés). [27]

Será necesario utilizar para el desarrollo de la aplicación, Apache, que es el servidor web más utilizado. Apache es un proyecto de código abierto y uso gratuito, multiplataforma, muy robusto y que destaca por su seguridad y rendimiento.

Su misión es crítica, ya que es el encargado de aceptar las peticiones de páginas (o recursos en general) que provienen de los visitantes que acceden a nuestro sitio web y gestionar su entrega o denegación, de acuerdo a las políticas

de seguridad establecidas. Esto, que puede parecer simple, implica muchas facetas y funcionalidades que debe cubrir, como pueden ser:

- Atender de manera eficiente, ya que puede recibir un gran número de peticiones HTTP, incluyendo una ejecución multitarea ya que pueden darse peticiones simultáneas. Cualquier petición compleja (por ejemplo, con acceso a base de datos) dejaría colapsado el servicio.
- Restricciones de acceso a los ficheros que no se quieran 'exponer', gestión de autenticaciones de usuarios o filtrado de peticiones según el origen de éstas.
- Manejar los errores por páginas no encontradas, informando al visitante y/o redirigiendo a páginas predeterminadas.
- Gestión de la información a transmitir en función de su formato e informar adecuadamente al navegador que está solicitando dicho recurso.
- Almacenar las peticiones recibidas, errores que se han producido y en general toda aquella información que puede ser registrada y analizada posteriormente para obtener las estadísticas de acceso al sitio web. [28]

Conclusiones Parciales:

Una vez terminado el estudio de distintos STI, tanto internacionales como nacionales, y de las metodologías, herramientas y lenguajes necesarios para el desarrollo del sistema. Analizadas las diferentes técnicas de IA para desarrollar el sistema, se arriba a las siguientes conclusiones:

- Los STI estudiados no cumplen con la arquitectura propuesta que propone la Dra. Zulma Cataldi.
- Las herramientas similares estudiadas no dan solución al problema planteado, dado que no existe ningún sistema tutorial inteligente para la asignatura de Reconocimiento de Patrones.
- Se seleccionó como técnica de IA, los Sistemas Basados en Reglas.
- La metodología ágil XP es adecuada para el desarrollo del software, ya que se adapta a las necesidades del mismo por las características del sistema a implementar.

Capítulo II. Propuesta de solución. Exploración y Planificación.

Introducción

En este capítulo se abordarán los aspectos fundamentales relacionados con el diseño del sistema a desarrollar. Entre los elementos a destacar se encuentran los módulos que componen el sistema y como se utilizan. Como vía para definir las futuras funcionalidades se generaron los artefactos relacionados a la especificación de las historias de usuarios y requisitos no funcionales del software.

2.1. Propuesta de solución

Se propone el desarrollo de un STI de apoyo a la asignatura de Reconocimiento de Patrones, este sistema está compuesto por los tres módulos definidos por la Dra. Zulma Cataldi: módulo dominio, módulo estudiante, módulo tutor y además la interfaz de usuario. Ver figura 7.

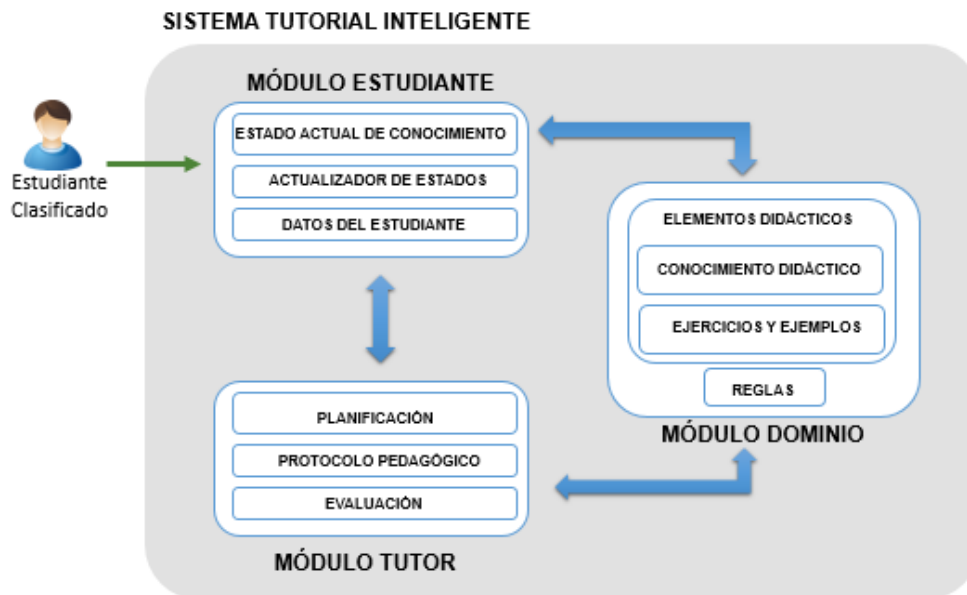


Figura 7. Propuesta de solución del Sistema. Fuente: Elaboración propia.

Como se evidencia en la figura anterior el sistema recibe a un estudiante clasificado según el estilo de aprendizaje y el tipo de inteligencia. A continuación, este realiza un test de nivel que permite conocer el dominio que presenta sobre la temática de la asignatura que debe vencer. Las evaluaciones de los diferentes

test de nivel fueron propuestas por los profesores de la asignatura RP, las cuales están divididas en bien (B), regular (R) y mal (M), según el por ciento de aciertos en las respuestas a las preguntas. Si el rango de respuesta está entre 0% – 59% se evalúa como M, si está entre 60% – 89% es R y 90% o más es B.

En dependencia de la clasificación inicial del estudiante y la evaluación obtenida en el test de nivel se le asigna un protocolo pedagógico. Los protocolos fueron creados previamente por el profesor de la asignatura, según las características anteriormente descritas. La técnica de IA Sistemas Basados en Reglas fue la seleccionada para la asignación de los protocolos pedagógicos. Siendo el antecedente: la clasificación de cada estudiante y la evaluación obtenida en el test de conocimiento y el consecuente: el protocolo pedagógico correspondiente a esos objetivos.

Ejemplo: **Si estudiante** pertenece a Clase1 (Estilo de aprendizaje/Tipo de Inteligencia) y Evaluación (X) **Entonces asigna** Protocolo X

Descripción de los módulos:

El **Módulo Dominio** será el encargado de tener almacenado todo el contenido que debe manejar el STI. Se encuentra estructurado por los siguientes sub-módulos: Parámetros Básicos del Sistema, Conocimientos y Elementos Didácticos.

- Conocimientos: es el sub-módulo que se encarga de asignar los protocolos pedagógicos a cada estudiante, guiándose por las reglas que tiene el sistema implementado. Tanto las reglas que se crearon para asignar la evaluación de los estudiantes como la de los protocolos. Además del conocimiento que genere con el motor de inferencia.
- Elementos Didácticos: es el sub-módulo que se encarga de gestionar toda la bibliografía que necesita el profesor para construir los protocolos pedagógicos. La bibliografía que se sube al sistema es almacenada en la base de datos, se clasifica según el tipo de bibliografía (imagen, videos y documentos) para que el profesor pueda conformar los protocolos pedagógicos de forma correcta y al tema al que pertenece cada bibliografía.

El **Módulo del Tutor** se encarga de definir y aplicar una estrategia pedagógica de enseñanza, que contiene los objetivos a ser alcanzados y selecciona el

material de aprendizaje para el estudiante. Se encuentra estructurado por los siguientes sub-módulos: Planificación, Protocolo Pedagógico y Evaluación.

- **Planificación:** es el sub-módulo que le permite al profesor gestionar todo el proceso de selección de las diferentes bibliografías que tiene el sistema, para utilizarla en el protocolo pedagógico. Este proceso se pone de manifiesto en la vista protocolo que es la que permite crear, editar, eliminar el protocolo pedagógico.
- **Protocolo Pedagógico:** es el sub-módulo encargado de gestionar todo el proceso de los protocolos: crearlos, editarlos, clasificarlos y eliminarlos. Todo esto se realiza en la clase protocolo la cual fue diseñada para que el profesor construya el protocolo pedagógico según las necesidades pedagógicas de los estudiantes. En esta sección es donde se crea el protocolo pedagógico y se selecciona la bibliografía que está en el sistema y que cumple con la misma clasificación del protocolo.
- **Evaluación:** es el sub-módulo que se encarga de la gestión del test de nivel. Donde está desarrollado todo el proceso de evaluar a los estudiantes, este proceso está formado por dos partes. Inicialmente cuando el estudiante trabaja por primera vez se le hace un test para ver el nivel de dominio de la asignatura y asignarle un protocolo pedagógico. Posteriormente que termine de consultar toda la bibliografía asignada en el protocolo, realiza otro test y según su calificación (B, R y M), se compara con la calificación anterior y se le asigna un nuevo protocolo.

El **Módulo Estudiante** se encarga del diagnóstico cognitivo del estudiante y el modelado del mismo. Se encuentra estructurado por los siguientes sub-módulos: Estilos de aprendizaje y Estado del conocimiento.

- **Estilos de aprendizaje:** el STI recibe ya clasificado a los estudiantes según sus diferentes estilos de aprendizaje, en la Tabla 1 se muestran los estilos de aprendizaje y sus descripciones. El sistema utiliza esa clasificación para asignarle a los estudiantes sus respectivos protocolos pedagógicos.

Tabla 1. Descripción de los Estilos de Aprendizaje. Fuente: Ing. Carlos E. Lemus Serrano. Estrategia metodológica para el desarrollo de un sistema tutor inteligente: módulo del dominio

Estilos de Aprendizaje	Descripción
Activo	Prefieren el desarrollo de ejercicios que leer. Les gusta el trabajo en grupos.
Reflexivo	Trabajan en grupo y prefieren la teoría. Aprenden mejor observando y reflexionando.
Teórico	Enfoque lógico de los problemas, necesitan integrar la experiencia en un marco teórico de referencia.
Visual	Prefieren imágenes, gráficos y representaciones visuales por lo que recuerdan mejor lo que ven.
Verbal	Prefieren explicaciones escritas o habladas.

- Estado del conocimiento: en este sub-módulo es donde se llevará un registro constante del progreso del estudiante, según la cantidad de ejercicios realizados, sus respectivas evaluaciones y los protocolos pedagógico que ha consultado.

Para desarrollar un correcto SBR fue necesario comprender la arquitectura por la que está compuesta dicha técnica de IA.

2.2. Diseño del Sistema Basado en Reglas

La arquitectura de los Sistemas Basados en Reglas está conformada por los siguientes componentes: motor de inferencia, base de datos, base de conocimiento e Interfaz. La relación de estos componentes se muestra en la Figura 8.

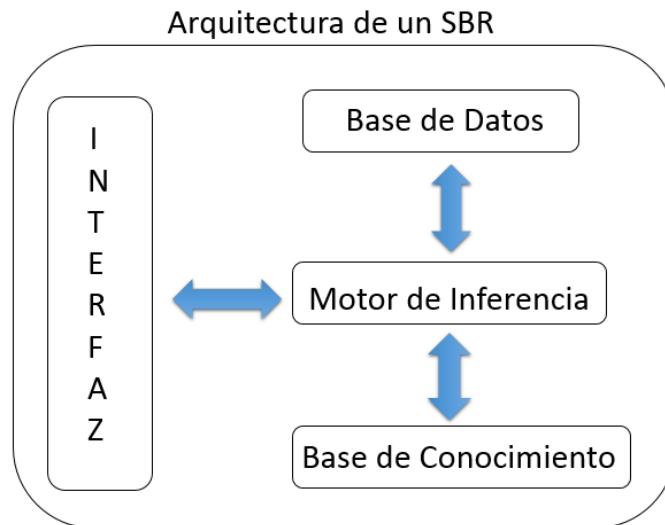


Figura 8. Propuesta de solución del Sistema. Fuente: Elaboración propia.

Base de Datos:

Contiene los datos sobre un problema que se han descubierto durante una consulta. Durante una consulta con el sistema experto, el usuario introduce la información del problema actual en la base de datos. El sistema empareja esta información con el conocimiento disponible en la base de casos para deducir los nuevos datos. La base de datos pertenece al módulo estudiante, es el que almacena toda la información que se posee del estudiante.

Base de Conocimiento:

Es la parte del sistema experto que contiene el conocimiento sobre el dominio. Hay que obtener el conocimiento del experto y codificarlo en la base de conocimiento. Una forma clásica de representar el conocimiento en un sistema experto son las reglas. La base de conocimiento pertenece al módulo dominio que es donde se encuentran todas las reglas y bibliografías del sistema.

Interfaz:

La interacción entre un sistema experto y un usuario se realiza en lenguaje natural. Es la que se encarga de recibir la información necesaria del usuario para el sistema.

Motor de Inferencia:

Trabaja con la información contenida en la base de conocimiento y la base de datos para deducir nuevos conocimientos. Contrasta los datos particulares de la base de datos con el conocimiento contenido en la base de conocimiento para obtener conclusiones acerca del problema. El motor de inferencia pertenece al módulo tutor, este módulo se encarga de asignarle el protocolo pedagógico al estudiante. A partir de los hechos, que son las características del estudiante almacenadas en el módulo estudiante y utiliza la base de conocimiento, que son las reglas las cuales se encuentran almacenadas en el módulo dominio.

Existen dos estructuras básicas para organizar el conocimiento de un SBR: las redes de inferencia y los sistemas de Patter-Matching. Para la implementación del STI, se seleccionó los sistemas Patter-Matching.

Estos sistemas realizan la búsqueda extensiva para “machear” y ejecutar las reglas, derivando nuevos hechos. Las relaciones entre las reglas y los hechos se forman en tiempo de ejecución.

Un sistema de Pattern-Matching depende del matching (compatibilidad) de las premisas de una regla con los hechos existentes para determinar cuáles reglas tienen sus premisas satisfechas por los hechos y, por eso, pueden ser ejecutadas.

Los SBR que usan Pattern-Matching son extremadamente flexibles y poderosos. Ellos son más aplicables a dominios donde hay muchas soluciones posibles. En estos dominios (diseños, planificación, etc.) no hay relaciones predeterminadas entre las reglas y los hechos.

El proceso de chequear las reglas para ver si ellas son satisfechas se denomina interpretación de las reglas.

La interpretación de las reglas en un motor de inferencia con razonamiento forward comprende los pasos básicos siguientes:

- Matching: se chequean cuales reglas son satisfechas.
- Resolución de conflictos: si hay varias reglas satisfechas es necesario seleccionar una.

- **Ejecución:** es ejecutar la regla seleccionada. De esta ejecución puede resultar que se añada un nuevo hecho (o hechos) a la base de datos o una nueva regla (o reglas) a la BC. [29]

A continuación, se muestra el algoritmo para la inferencia en pseudocódigo:

1. Identificar el conjunto S de reglas aplicables.
2. Mientras S no sea vacío,
 - a) Seleccionar una regla R de S.
 - b) Aplicar R, generando los nuevos estados y añadiéndolos a la B.D.
 - c) Si se generó el estado objetivo entonces TERMINAR y ÉXITO.
 - d) Si no, llamar nuevamente al procedimiento generar.
 - e) Eliminar R de S y anular el efecto de aplicar R.

2.3. Usuarios relacionados con el sistema

A continuación se muestra en la Tabla 2, una descripción general que realizan los diferentes usuarios en el sistema.

Tabla 2. Usuarios relacionados con el sistema

Usuarios relacionados con el sistema	Argumentación
Estudiante	Es el usuario al que se le asignan los protocolos pedagógicos.
Profesor	Puede gestionar cualquiera de los protocolos pedagógicos y los reportes sobre los estudiantes.
Administrador	Es el usuario encargado de administrar el sistema, que consiste en gestionar las cuentas de los usuarios y el contenido publicado en el sistema.

2.4. Fase de exploración

En esta fase se precisan las historias de usuario, determinando su prioridad en correspondencia con la importancia que posee para el desarrollo del sistema

2.4.1. Historias de Usuario

Las HU son el primer paso de cualquier proyecto que siga la metodología XP. Son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. El tiempo ideal para su desarrollo, es entre 1 y 3 semanas.

Estructuras de las Historias de Usuarios:

Las HU están compuestas por varias secciones:

- **Nombre:** nombre que identifica la HU.
- **Número:** número de la HU.
- **Usuario:** nombre de usuarios que interacciona.
- **Prioridad en el Negocio:**
 - alta:** funcionalidades fundamentales en el desarrollo de la aplicación.
 - media:** funcionalidades a tener en cuenta pero que no tienen unan afectación sobre el sistema.
 - baja:** sirven de ayuda a los desarrolladores, pero no tienen nada que ver con el sistema.
- **Estimación:** tiempo estimado que se demorarán en implementar los desarrolladores en implementar las HU en días.
- **Descripción:** breve descripción de la funcionalidad que realizará la HU.

A continuación, se muestran el nombre de cada una de las historias de usuario del sistema y se muestra la descripción de la HU Insertar protocolo. ([Ver Anexo 1, Historias de Usuarios](#))

- HU 1: Autenticar Usuario.
- HU 2: Insertar Usuario.
- HU 3: Modificar Usuario.
- HU 4: Listar Usuario.
- HU 5: Eliminar Usuario.
- HU 6: Insertar estudiante
- HU 7: Listar estudiante.
- HU 8: Buscar datos del estudiante.

- HU 9: Insertar Bibliografía.
- HU 10: Mostrar Bibliografía.
- HU 11: Eliminar Bibliografía.
- HU 12: Insertar Cuestionario.
- HU 13: Mostrar Cuestionario
- HU 14: Eliminar Cuestionario.
- HU 15: Insertar protocolo pedagógico.
- HU 16: Mostrar Protocolo Pedagógico.
- HU 17: Eliminar Protocolo Pedagógico.
- HU 18: Modificar Protocolo Pedagógico.

Tabla 3. HU: Insertar protocolo pedagógico

Historia de Usuario	
Nombre: Insertar protocolo pedagógico	
Número: 15	Usuario: Profesor
Prioridad: Alta	Puntos Estimados: 8
Descripción: El profesor crea un nuevo protocolo pedagógico con la bibliografía existente en el sistema.	

2.4.2. Requisitos no funcionales del sistema

Propiedades o cualidades que el producto debe tener. Representan las características del producto.

Software

- Se requiere que la aplicación sea multiplataforma.
- **Requerimientos para el servidor:** Es necesario la instalación del gestor de base de datos phpMyAdmin 4.1.14, el servidor web Apache 2.4.9, PHP 5.5.12 y el servidor mysql 5.7.14.

Hardware para el Servidor:

La PC donde se ejecutará la aplicación debe tener las siguientes prestaciones mínimas, 1GB de RAM y un microprocesador CELEROM con velocidad de 2.5 GHz.

Seguridad

La aplicación cuenta con un proceso de autenticación de usuarios, el cual funciona como aspecto de seguridad, donde cada usuario tiene asignado su rol, además se utiliza un algoritmo de encriptación MD5 para proteger las contraseñas.

2.5. Fase de planificación

La planificación es una fase corta, en la que el cliente y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario. El resultado de esta fase es un Plan de Entregas o de Iteraciones.

2.5.1. Plan de iteraciones

Las HU son divididas en tareas de entre 1 y 3 días que son asignadas a los programadores, creándose el plan de duración de cada una de las iteraciones, así como el orden en que serán implementadas.

Tabla 4. Plan de Iteraciones

Iteraciones	Historias de Usuarios	Cantidad de tiempo de trabajo
1	Autenticar usuario Insertar usuario Modificar usuario Listar Usuario Eliminar Usuario	4 semanas
2	Insertar Estudiante Listar estudiantes Buscar datos del estudiante	5 semanas
3	Insertar bibliografía Mostrar bibliografía Eliminar bibliografía	5 semanas
4	Insertar cuestionario Mostrar cuestionario Eliminar cuestionario	5 semanas
5	Insertar protocolo pedagógico	6 semanas

	Mostrar protocolo pedagógico Eliminar protocolo pedagógico Modificar protocolo pedagógico Asignar protocolo pedagógico	
--	---	--

Conclusiones Parciales:

Con el desarrollo de las fases de: exploración y planificación propuestos por la metodología XP, detallando las Historias de Usuario y el tiempo estimado para la entrega de cada una podemos concluir :

- Se determinaron las características funcionales del sistema representadas en historias de usuario, y se planificaron las diferentes iteraciones permitiendo organizar el desarrollo del sistema.
- Se elaboró el plan de entregas, donde se estima que el software estará listo en aproximadamente 6 meses y 1 semana.

Capítulo III. Diseño, implementación y Prueba.

Introducción

Dentro del proceso de desarrollo de software, la etapa de implementación es la encargada de convertir una especificación del sistema en un sistema ejecutable. En la etapa de pruebas el resultado final, o sea el ejecutable, es evaluado en cuanto a su calidad y desempeño como producto de software.

En el capítulo se visualiza como se llevó a cabo la implementación del sistema, definiéndose diferentes tareas por iteraciones. El capítulo culmina con las pruebas que se realizaron en conjunto con el cliente, para definir si el sistema cumplía con los objetivos requeridos.

3.1. Fase de diseño

El papel del diseño en el ciclo de vida del software es adquirir una comprensión de su funcionamiento, además de crear una entrada apropiada o arquitectura de software, que constituya el punto de partida para las actividades de implementación.

3.1.1. Tarjetas CRC

La metodología XP sugiere diseños simples y sencillos para conseguir que sea fácil y entendible, logrando menos tiempo y esfuerzo a la hora de desarrollar. La utilización de tarjetas CRC (Class-Responsibility-Collaboration) es una técnica de diseño orientado a objetos. El objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar, esperando que el diseño sea lo más simple posible verificando las especificaciones del sistema. [30]

Las tarjetas CRC se dividen en tres secciones: **nombre de la clase**, las **responsabilidades** (lo que la clase sabe o hace) y **colaboradores** (otras clases con las que trabaja en conjunto para llevar a cabo sus funcionalidades)

A continuación, se muestra un ejemplo de una tarjeta CRC:

Tabla 5. Tarjeta CRC: Gestionar Usuario

Clase: Usuario	
Responsabilidades:	Colaboradores:
Autenticar usuario	conexión
Insertar usuario	
Modificar usuario	
Listar Usuario	
Eliminar usuario	

Tabla 6. Tarjeta CRC: Gestionar Estudiante

Clase: Estudiante	
Responsabilidades:	Colaboradores:
Insertar los estudiantes	conexión
Listar los estudiantes	
Buscar datos del estudiante	

Tabla 7. Tarjeta CRC: Gestionar Bibliografía

Clase: Bibliografía	
Responsabilidades:	Colaboradores:
Insertar bibliografía	conexión
Mostrar bibliografía	
Eliminar bibliografía	

Tabla 8. Tarjeta CRC: Gestionar Cuestionario

Clase: Cuestionario	
Responsabilidades:	Colaboradores:
Insertar cuestionario	conexión
Mostrar cuestionario	
Eliminar cuestionario	

Tabla 9. Tarjeta CRC: Gestionar Protocolo

Clase: Protocolo	
Responsabilidades:	Colaboradores:
Insertar protocolo pedagógico	conexión
Mostrar protocolo pedagógico	
Eliminar protocolo pedagógico	
Modificar protocolo pedagógico	
Asignar protocolo pedagógico	

3.1.2. Estilo Arquitectónico

Los patrones de la arquitectura de un sistema, definen la estructura fundamental con la que se va a desarrollar la aplicación. Para el desarrollo del STI se utilizó el Modelo-Vista-Controlador (MVC).

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar el modelo y la vista de los detalles del protocolo utilizado para las peticiones. El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes del tipo de gestor de bases de datos utilizado por la aplicación. [31]

- **Modelo**

Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio, así como acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. Por lo que se definen reglas de negocio (la funcionalidad del sistema), las cuales, pueden estar también en los controladores, directamente en las acciones. Notificará a las vistas los cambios que en los datos pueda producir un agente externo si se está ante un modelo activo. [31]

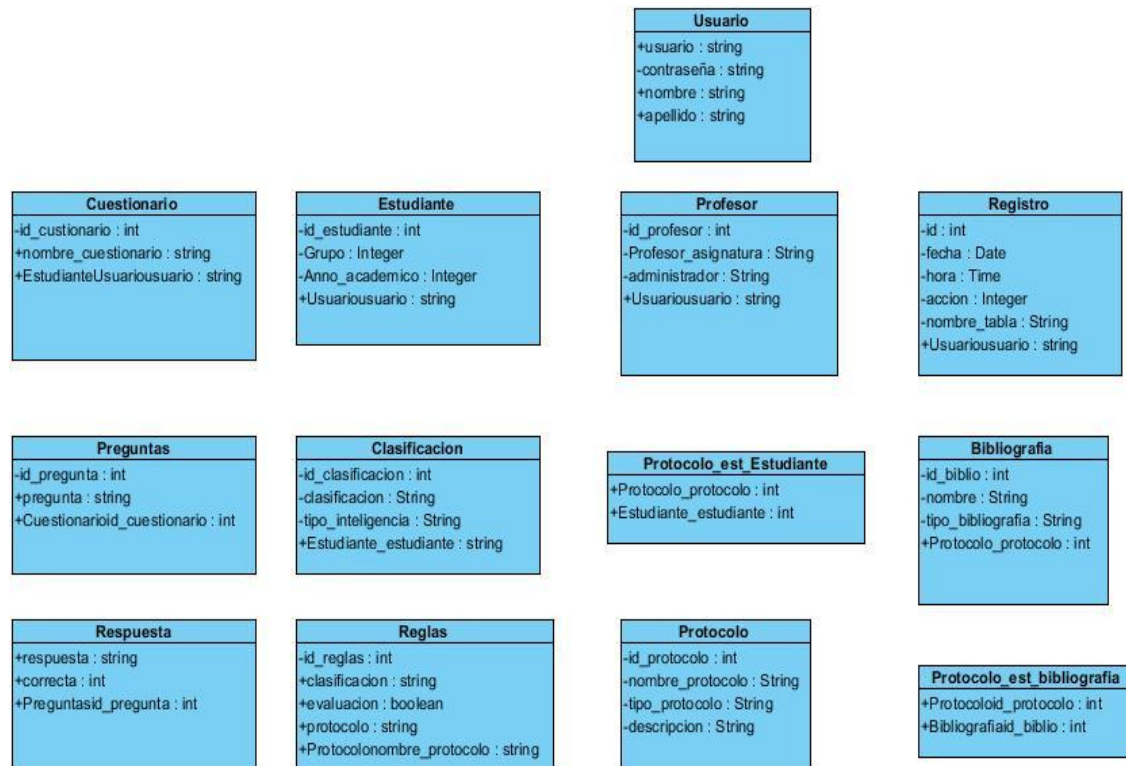


Figura 9. Vista del Modelo

- **Usuario:** clase encargada de registrar los usuarios.
- **Estudiante:** clase encargada de almacenar los datos de todos los estudiantes.
- **Profesor:** clase encargada de almacenar los datos de todos los profesores.
- **Registro:** clase encargada de almacenar las acciones realizadas por los profesores sobre el sistema.
- **Clasificación:** clase encargada de registrar la clasificación de cada estudiante, teniendo en cuenta sus características.
- **Protocolo:** clase encargada de registrar la información de los protocolos pedagógicos.
- **Protocolo_est_Estudiante:** clase encargada de registrar la relación entre las tablas protocolo y estudiante.
- **Bibliografía:** clase encargada de almacenar toda la bibliografía.
- **Protocolo_est_bibliografia:** clase encargada de registrar la relación entre las tablas protocolo y bibliografía.
- **Cuestionario:** clase encargada de registrar todos los cuestionarios que se crean en el sistema.

- **Preguntas:** clase encargada de registrar todas las preguntas que se crean para los cuestionarios del sistema.
- **Respuesta:** clase encargada de registrar todas las respuestas de las preguntas que se realizan en el cuestionario.
- **Reglas:** clase encargada de registrar todas las reglas del sistema para poder asignar los protocolos pedagógicos.

- **Vista**

Transforma el modelo en una página web, permite al usuario interactuar con ella. Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario. Tienen un registro de su controlador asociado. [31]

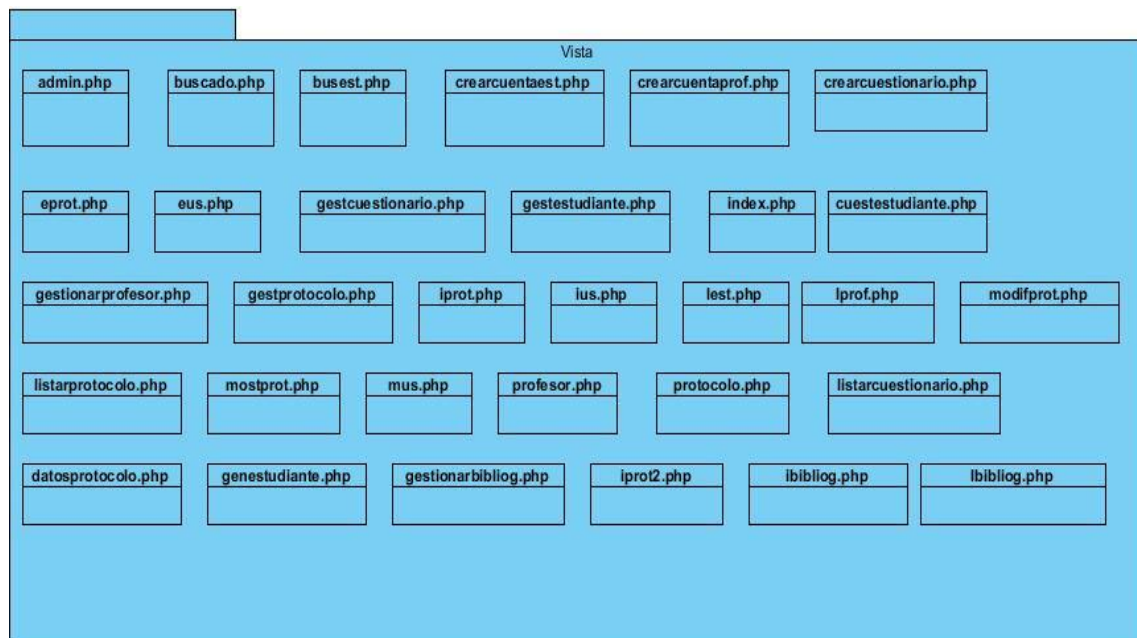


Figura 10. Vista de Vista

- **Controlador**

Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Recibir los eventos de entrada. Contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a los métodos. [31]

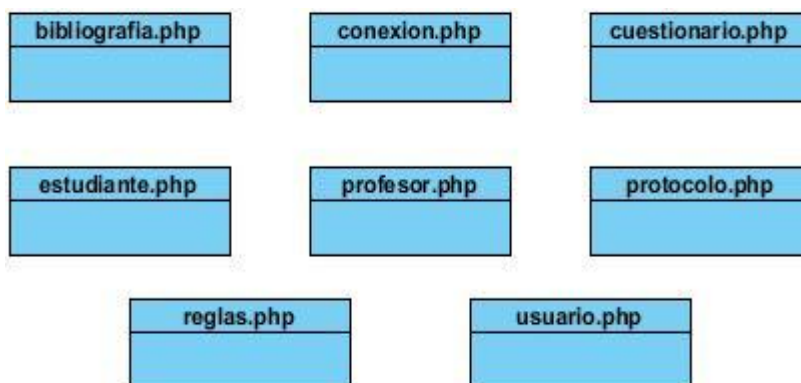


Figura 11. Vista del Controlador

A continuación, se describen las funciones de cada clase controladora.

- **conexion:** clase encargada de establecer la conexión con la base de datos.
- **estudiante:** clase encargada de controlar la gestión de los estudiantes.
- **profesor:** clase encargada de controlar la gestión de los profesores.
- **protocolo:** clase encargada de controlar la gestión de los protocolos.
- **usuario:** clase encargada de controlar la gestión de los usuarios.

3.1.3. Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Su utilización permite ahorrar grandes cantidades de tiempo en la construcción de software, siendo más fácil de comprender, mantener y extender.

Los patrones de Asignación de Responsabilidades (GRASP) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Dentro de los patrones GRASP que se utilizaron se encuentran:

- **Patrón Experto** es el encargado de asignar una responsabilidad al experto en información, no es más que la clase que cuenta con la información necesaria para cumplir la responsabilidad. Esto se evidencia en las clases modelos y las clases controladoras. Ejemplo, la clase controladora estudiante, será la encargada de insertar, modificar y eliminar cualquier estudiante.
- **Patrón Controlador** es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Un ejemplo de esto se evidencia en la clase usuario, con el método `crear_cuenta()`, el cual

3.3. Despliegue del sistema

El diagrama de despliegue es un diagrama del UML, se utiliza para modelar el hardware y muestra la disposición física de los distintos nodos que componen un sistema y las relaciones. El diagrama de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución como una computadora, un dispositivo, conectados por asociaciones de comunicación tales como enlaces de red, conexiones HTTP, etc. [33]

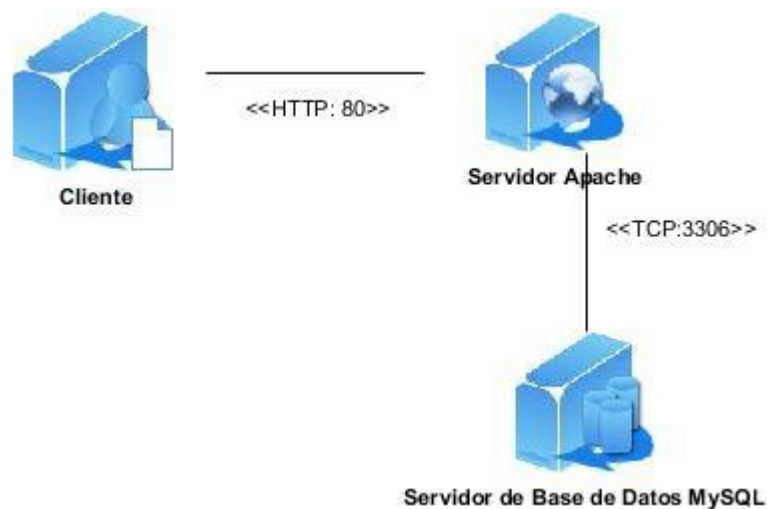


Figura 13. Diagrama de Despliegue

3.4. Tareas de Ingeniería

Durante la fase de implementación siguiendo la metodología XP, se realiza la implementación de 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. A continuación, se enumeran las tareas de ingeniería por cada HU en cada una de las iteraciones.

Iteración 1

El principal objetivo de esta iteración es desarrollar las HU 1. Autenticar usuario, 2. Insertar usuario, 3. Modificar usuario, 4 Listar usuario y 5. Eliminar usuario.

HU 1. Autenticar usuario se desarrollan las siguientes tareas:

Tarea 1: Comprueba que el usuario y la contraseña existan en la base de datos.

Tarea 2: Le asigna el rol según los permisos que tenga el usuario

HU 2. Insertar usuario se desarrollan las siguientes tareas:

Tarea 3: Comprobar que el usuario no esté creado en la base de datos.

Tarea 4: Guardar los datos en la base de datos.

HU 3. Modificar usuario se desarrollan las siguientes tareas:

Tarea 5: Buscar el usuario en la base de datos.

Tarea 6: Modificar el rol que ocupa el usuario en la base de datos.

HU 4. Listar usuario se desarrollan las siguientes tareas:

Tarea 7: Listar todos los usuarios del sistema

HU 5. Eliminar usuario se desarrollan las siguientes tareas:

Tarea 8: Buscar el usuario en la base de datos.

Tarea 9: Eliminar el usuario de la base de datos.

Iteración 2

El principal objetivo de esta iteración es desarrollar las HU 6. Insertar Estudiante, 7. Listar Estudiante y 8. Buscar Datos del Estudiante.

HU 6. Insertar Estudiante se desarrollan las siguientes tareas:

Tarea 10: Comprobar que el estudiante no exista en la base de datos.

Tarea 11: Guardar datos en la base de datos y mostrar sección correspondiente.

Tarea 12: Salir del sistema en cualquier momento.

HU 7. Listar Estudiante se desarrollan las siguientes tareas:

Tarea 13: Mostrar todos los datos de todos los estudiantes en la base de datos.

HU 8. Buscar Estudiante se desarrollan las siguientes tareas:

Tarea 14: Mostrar todos los estudiantes en la base de datos que cumplen con el criterio de búsqueda.

Iteración 3

El principal objetivo de esta iteración es desarrollar las HU 8. Insertar bibliografía, 9. Mostrar bibliografía y 10. Eliminar bibliografía.

HU 8. Insertar bibliografía se desarrollan las siguientes tareas:

Tarea 15: Seleccionar la bibliografía para subirla al sistema.

Tarea 16: Rellenar los campos de clasificación de la bibliografía.

HU 9. Mostrar bibliografía se desarrollan las siguientes tareas:

Tarea 17: Listar la bibliografía de la base de datos.

Tarea 18: Muestra toda la bibliografía existente en el sistema.

HU 10. Eliminar bibliografía se desarrollan las siguientes tareas:

Tarea 19: Eliminar la bibliografía de la base de datos.

Iteración 4

El principal objetivo de esta iteración es desarrollar las HU 11. Insertar cuestionario, 12. Mostrar cuestionario y 13. Eliminar cuestionario.

HU 11. Insertar cuestionario se desarrollan las siguientes tareas:

Tarea 20: Buscar que el cuestionario no esté creado en la base de datos.

Tarea 21: Crear el nuevo cuestionario en la base de datos.

HU 12. Mostrar cuestionario se desarrollan las siguientes tareas:

Tarea 22: Listar todos los cuestionarios existentes en la base de datos.

HU 13. Eliminar cuestionario se desarrollan las siguientes tareas:

Tarea 23: Eliminar el cuestionario específico existente en la base de datos.

Iteración 5

El principal objetivo de esta iteración es desarrollar las HU 14. Insertar protocolo, 15. Mostrar protocolo, 16. Eliminar protocolo, 17. Modificar protocolo y 18. Asignar protocolo.

HU 14. Insertar protocolo se desarrollan las siguientes tareas:

Tarea 24: Listar la bibliografía de la base de datos.

Tarea 25: Insertar la bibliografía necesaria para ese protocolo.

Tarea 26: Buscar que el protocolo no este creado.

HU 15. Mostrar protocolo se desarrollan las siguientes tareas:

Tarea 27: Lista todos los protocolos existentes en la base de datos.

HU 16. Eliminar protocolo se desarrollan las siguientes tareas:

Tarea 28: Eliminar protocolo específico existente en la base de datos.

HU 17. Modificar protocolo se desarrollan las siguientes tareas:

Tarea 29: Comprobar que el protocolo esté creado en la base de datos.

Tarea 30: Modificar los datos en la base de datos.

HU 18. Asignar protocolo se desarrollan las siguientes tareas:

Tarea 31: Asigna el protocolo correspondiente al estudiante.

Tarea 32: Crea una regla nueva y le asigna el protocolo correspondiente.

A continuación, se muestra la descripción de la Tarea de Ingeniería 13 Listar la bibliografía de la base de datos, el resto de las descripciones se encuentran en los anexos. ([Ver Anexo 2, Tareas de Ingeniería.](#))

Tabla 10. Tarea de Ingeniería 13. Listar la bibliografía de la base de datos

Tarea de Ingeniería	
Número de Tarea: 19	Número de HU: 11
Nombre de la Tarea: Eliminar la bibliografía de la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 7 día
Fecha de Inicio: 25/03/2017	Fecha Fin: 31/03/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: listar toda la bibliografía existente en el sistema.	

3.5. Pruebas

Para el STI las pruebas unitarias fueron desarrolladas constantemente una vez terminada de implementar cada funcionalidad cumpliendo con los requisitos planteados.

Las pruebas de aceptación son creadas en base a las HU en cada ciclo de la iteración del desarrollo, donde una HU puede tener todas las pruebas de aceptación que se necesite para asegurar su correcto funcionamiento. El cliente debe especificar uno o diversos escenarios para comprobar que la HU ha sido correctamente implementada y en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una HU no se puede considerar terminada hasta que no pase correctamente todas las pruebas de aceptación.

Estas pruebas son representadas mediante tablas divididas por las siguientes secciones:

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 muestra la descripción del caso de prueba de aceptación Insertar Protocolo Pedagógico, el resto de las descripciones se encuentran en los anexos. ([Ver Anexo 3, Ejemplos de Casos de pruebas de aceptación.](#))

Tabla 11. Caso de prueba de aceptación HU14_P1. Insertar Protocolo

Caso de prueba de aceptación	
Código: HU15_P1	Historia de Usuario: 15
Nombre: Insertar Protocolo Pedagógico	
Descripción: prueba para la funcionalidad Insertar Protocolo Pedagógico	
Condiciones de Ejecución: el usuario selecciona la bibliografía que se le asignará, Inserta el nombre, el tipo y la descripción del protocolo.	
Entrada/ Pasos de Ejecución: el usuario selecciona la bibliografía que se le asignará, Inserta el nombre, el tipo, la descripción del protocolo y oprime el botón insertar. El sistema verifica que el protocolo no esté creado y lo inserta en la base de datos.	
Resultado Esperado: el protocolo pedagógico es insertado en la base de datos.	
Evaluación de la Prueba: Prueba Satisfactoria	

3.6. Resultado de las pruebas realizadas

La realización de pruebas al sistema permitió detectar varias no conformidades en las cinco iteraciones, siendo estas resueltas. Concluida esta fase de pruebas, fue posible comprobar que el sistema cumple con las especificaciones que se realizaron en las HU. A continuación, se muestra la relación de no conformidades por iteración:

Tabla 12. Resultado de las pruebas

Iteración	NC Detectadas	Asociadas a	NC Resueltas
1	4	Errores de interfaz Errores ortográficos	3
2	2	Errores de interfaz	2
3	1	Errores ortográficos	1
4	4	Errores de interfaz	3
5	5	Errores de interfaz Errores ortográficos	5

Como se puede apreciar en la tabla anterior se realizaron cinco iteraciones de pruebas. A lo largo de la primera iteración se detectaron cuatro no conformidades, tres asociadas a errores de interfaz y una a errores ortográficos, tres fueron resueltas en la propia iteración. En la segunda iteración fueron detectadas dos no conformidades asociadas a errores de interfaz siendo satisfactoriamente solucionadas. En la tercera iteración se detectó una no conformidad asociada a errores ortográficos, resuelta en la propia iteración. En la cuarta iteración fueron detectadas cuatro no conformidades asociadas a errores de interfaz siendo solucionadas tres. En la quinta iteración se detectaron cinco no conformidades, tres asociadas a errores de interfaz y dos a errores ortográficos siendo satisfactoriamente solucionadas por lo que la aplicación mostró un buen funcionamiento y se considera terminada.

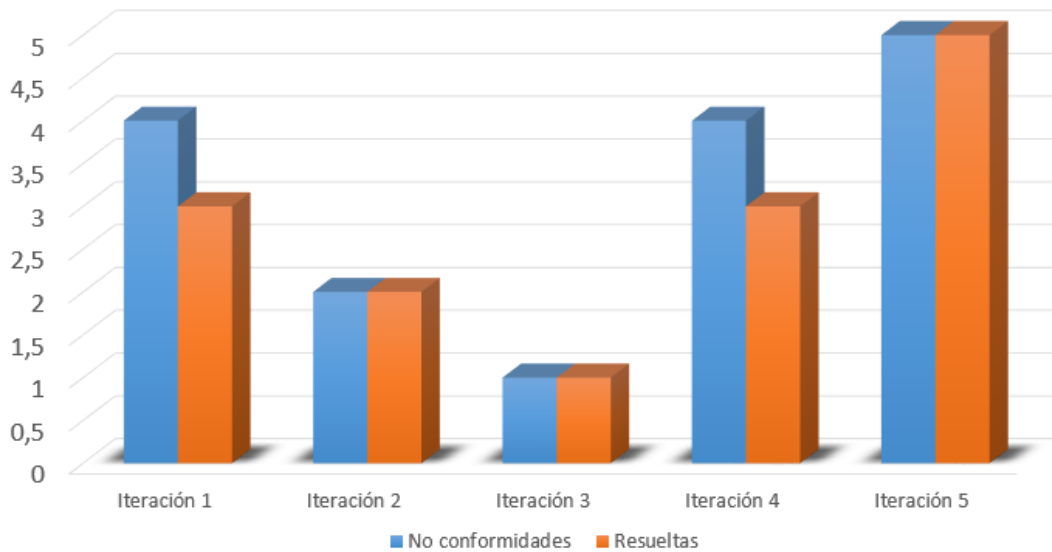


Figura 14. Pruebas de Aceptación

Conclusiones Parciales:

En este capítulo se abordó la etapa de diseño, implementación y prueba del Sistema Tutorial Inteligente para la asignatura de Reconocimiento de Patrones, describiendo la arquitectura MVC que se utiliza, identificando los patrones de diseño, así como las tareas de ingeniería necesarias por cada HU, el diagrama de despliegue del mismo y las pruebas realizadas. Se arriban a las siguientes conclusiones:

- Se eligió la arquitectura de software orientada a objetos por sus características determinantes en la implementación del sistema inteligente y el uso de patrones de diseño que contribuyen a la implementación de buenas prácticas en el desarrollo del sistema.
- Las pruebas unitarias realizadas en cada iteración del sistema mostraron los errores encontrados en las funcionalidades facilitando así la detección de los mismos en la implementación y dándole solución.
- Las pruebas de aceptación permitieron verificar que el sistema realiza correctamente los requisitos solicitados por el cliente.

Conclusiones

En el presente trabajo se logró el desarrollo de una aplicación web, estructurada en tres capítulos como dicta la metodología de desarrollo de software que se seleccionó. Una vez completada la investigación, se puede concluir:

- Se desarrolló una aplicación web, capaz de ayudar a los estudiantes de la asignatura de Reconocimiento de Patrones a consolidar su conocimiento y lograr aprovechar su tiempo de auto estudio, mostrándoles la bibliografía más acorde a sus características individuales.
- Se determinó cómo están estructurados los STI y las diferentes técnicas de IA que se utilizan para el desarrollo de estas herramientas educativas. Lo que permitió la selección adecuada de la técnica de IA a utilizar, así como los elementos a tener en cuenta para la propuesta de solución.
- La evaluación de las pruebas de software realizadas permitió suprimir las no conformidades detectadas en la aplicación desarrollada.

Recomendaciones

Una vez concluida la investigación y el desarrollo de la propuesta de solución, los autores del presente trabajo recomiendan:

- Implementar nuevos formatos de preguntas para la elaboración de los cuestionarios evaluativos.
- Desarrollar una funcionalidad estadística que permita controlar la evolución del estudiante.

Referencias Bibliográficas

1. Importancia de las TIC en el proceso de enseñanza aprendizaje. [online]. Available from: <http://www.educando.edu.do/articulos/docente/importancia-de-las-tic-en-el-proceso-de-enseanza-aprendizaje/>
2. **Marilyn Sandra Choque Callisaya.** Técnicas de la Inteligencia Artificial Aplicadas a la Educación. 2008 [online]. Available from: www.revistasbolivianas.org.bo/pdf/rits/n1/n1a10.pdf
3. Plataforma Educativa ZERA. [online]. Available from: eva.uci.cu/es/aboutAs.
4. Los recursos de aprendizaje en la educación a distancia Nuevos escenarios,... -. [online]. [Accessed 9 June 2017]. Available from: <http://www.ebah.com.br/content/ABAAAgw6EAI/los-recursos-aprendizaje-en-la-educacion-a-distancia-nuevos-escenarios-experiencias-y-tendencias>
Arquivo Livro Peru.pdf
5. Ventajas y Desventajas de las Plataformas Virtuales - Plataformas Educativas. [online]. [Accessed 9 June 2017]. Available from: <https://sites.google.com/site/plataformaseducativasvirtuales/home/plataformas-virtuales/ventajasdesventajas>
6. **Fernando J. Lage, Zulma Cataldi.** Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia [online]. Facultad Regional Buenos Aires. Universidad Tecnológica Nacional. Available from: www.virtualeduca.info/ponencias2009/91/Modelo%20STI%20Internet.doc
7. **Fernando Salgueiro, Zulma Cataldi and Ramón García–Martínez.** *Los estilos pedagógicos en el modelado del tutor para Sistemas Tutores Inteligentes.* [online]. Facultad de Ingeniería. Universidad de Buenos Aires. Available from: laboratorios.fi.uba.ar/lie/Revista/Articulos/020204/A4mar2005.pdf
8. **Nestor D. Duque, Héctor M. González and Demetrio A. Ovalle.** Modelo del Estudiante para Sistemas Adaptativos de Educación Virtual. [online]. May 2008. Revista Avances en Sistemas e Informática. Available from: <http://www.redalyc.org/articulo.oa?id=133114993004>

9. **Enrique Sucar Succar, Mario A. Romero Inzunza and Pilar Gómez-Gil.** Diseño de HEDEA: una herramienta para la construcción de Sistemas Tutores Inteligentes. Noviembre 2009 [online]. Available from: <https://ccc.inaoep.mx/~pgomez/publications/congress/MriInR09.pdf>
10. **Yaneidis Hernández Orozco.** Los Sistemas Tutores Inteligentes aplicados en la Educación. *Revista TINO ISSN 1995-9419* [online]. 30 May 2016. [Accessed 9 June 2017]. Available from: <https://revista.jovenclub.cu/los-sistemas-tutores-inteligentes-aplicados-en-la-educacion>
11. **Luisa Idorka Morales Rodríguez and Yarisleydi De Avila Fernández.** *Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos.* Universidad de las Ciencias Informáticas, 2012.
12. **Anita Ferreira, Pedro Salcedo.** La Arquitectura de ELE-TUTOR: Un Sistema Tutorial Inteligente para el Español como Lengua Extranjera. [online]. Available from: www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-09342012000200001
13. **María Lucía Barrón Estrada, Ramón Zatarain Cabada and Yasmín Hernández Pérez.** Tutor Inteligente con reconocimiento y manejo de emociones para Matemáticas *Revista Electrónica de Investigación Educativa.* . 2014. Vol. Vol. 16, no. Núm. 3.
14. **Ricardo Rosas, Rodrigo Zamorano, Miguel Nussbaum, Rodrigo Buzeta and Manuela Moulian, Jaime Bermeosolo.** Desarrollo y Evaluación de un Sistema Tutorial Inteligente para el Apoyo de la Enseñanza de la Lectura Inicial. [online]. Vol. Vol 6, no. 1. Available from: <http://www.psykhe.cl/index.php/psykhe/article/view/102>
15. **Lidiana Cruz Pazos, José Enrique Pacheco Piñeyro.** Sistema de enseñanza-aprendizaje inteligente de apoyo al curso introductorio de Matemática. Universidad de las Ciencias Informáticas, [no date].
16. **Laura Casas Fuentes, Olga Lidia Pérez González, Lisandra Docampo, Yailé Caballero Mota, Lenniet Coello, Isabel Yordi González and Angela**

Martín. SISTEMA INTELIGENTE PARA EL ALGEBRA LINEAL. Facultad de Informática, Univerisdad de Camagüey y Universidad APEC, [no date].

17. **Verónica Amela Tarongí.** Sistema Tutor Inteligente Adaptativo para Laboratorios Virtuales y Remotos [online]. Valencia : Universidad Politécnica de Valencia, Departamento de Ingeniería de Sistemas y Automática, 2010. Available from: <https://riunet.upv.es/bitstream/handle/10251/13851/Trabajo%20de%20Investigacion.pdf?sequence=1>
18. **José Manuel Gutiérrez.** Sistemas Expertos Basados en Reglas [online]. Dpto. de Matemática Aplicada. Universidad de Cantabria. Available from: <http://personales.unican.es/gutierjm/cursos/expertos/Reglas.pdf>
19. **Maykel Leyva.** Aplicación de la estrella Boehm turner a los proyectos de desarrollo de software en la Universidad de Guayaquil. [online]. Available from: <https://www.authorea.com/users/110721/articles/142954-aplicaci%C3%B3n-de-la-estrella-boehm-turner-a-los-proyectos-de-desarrollo-de-software-en-la-universidad-de-guayaquil>
20. **Ing. José Joskowicz.** Reglas y Prácticas en eXtreme Programming. [online]. Available from: <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
21. Qué es UML? | LENGUAJE DE MODELADO UNIFICADO UML. [online]. [Accessed 8 June 2017]. Available from: http://stadium.unad.edu.co/ovas/10596_9839/qu_es_uml.html
22. ¿Qué es PHP? [online]. Available from: <http://php.net/manual/es/intro-what-is.php>.
23. **Javier Eguíluz Pérez.** Introducción a javascript. [online]. Available from: www.librosweb.es/javascript.
24. MySQL | La base de datos de código abierto más popular | Oracle América Latina. [online]. [Accessed 9 June 2017]. Available from: <https://www.oracle.com/lad/mysql/index.html>

25. Introducción a Herramientas CASE Y System Architect. [online]. Available from: http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf.
26. Visual Paradigm Para UML. [online]. Available from: <http://www.software.com.ar/p/visual-paradigm-para-uml#product-description>.
27. Apache HTTP Server: ¿Qué Es, Cómo Funciona Y Para Qué Sirve? [online]. 2014. Available from: <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.
28. ¿Qué Hace Un Servidor Web Como Apache? [online]. 2012. Available from: <http://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>.
29. **Dr. Daniel Gálvez Lio.** Curso de Sistemas Basados en el Conocimientos. Especializa en Inteligencias Artificial. Grupo de Investigación en Inteligencia Artificial Departamento de Ciencia de la Computación Facultad de Matemática, Física y Computación Universidad Central “Martha Abreu” de Las Villas. 1998.
30. Desarrollo de Software. Tarjetas CRC. [online]. 2012. Available from: <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>.
31. **Yunisleidy Fernández Romero.** Patrón modelo-Vista-Controlador.
32. **I. Sommerville.** Software Engineering.
33. **Patricia López, Francisco Ruiz.** Lenguaje Unificado de Modelado - UML. Universidad Cantabria – Facultad de Ciencias.

Bibliografía

1. Importancia de las TIC en el proceso de enseñanza aprendizaje. [online]. Available from: <http://www.educando.edu.do/articulos/docente/importancia-de-las-tic-en-el-proceso-de-enseanza-aprendizaje/>
2. **Marilin Sandra Choque Callisaya.** Técnicas de la Inteligencia Artificial Aplicadas a la Educación. 2008 [online]. Available from: www.revistasbolivianas.org.bo/pdf/rits/n1/n1a10.pdf
3. Plataforma Educativa ZERA. [online]. Available from: eva.uci.cu/es/aboutAs.
4. Los recursos de aprendizaje en la educación a distancia Nuevos escenarios,... -. [online]. [Accessed 9 June 2017]. Available from: <http://www.ebah.com.br/content/ABAAAgw6EAl/los-recursos-aprendizaje-en-la-educacion-a-distancia-nuevos-escenarios-experiencias-y-tendencias>
Arquivo Livro Peru.pdf
5. Ventajas y Desventajas de las Plataformas Virtuales - Plataformas Educativas. [online]. [Accessed 9 June 2017]. Available from: <https://sites.google.com/site/plataformaseducativasvirtuales/home/plataformas-virtuales/ventajasdesventajas>
6. **Fernando J. Lage, Zulma Cataldi.** Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia [online]. Facultad Regional Buenos Aires. Universidad Tecnológica Nacional. Available from: www.virtualeduca.info/ponencias2009/91/Modelo%20STI%20Internet.doc
7. **Fernando Salgueiro, Zulma Cataldi and Ramón García–Martínez.** *Los estilos pedagógicos en el modelado del tutor para Sistemas Tutores Inteligentes.* [online]. Facultad de Ingeniería. Universidad de Buenos Aires. Available from: laboratorios.fi.uba.ar/lie/Revista/Articulos/020204/A4mar2005.pdf
8. **Nestor D. Duque, Héctor M. González and Demetrio A. Ovalle.** Modelo del Estudiante para Sistemas Adaptativos de Educación Virtual. [online]. May 2008. Revista Avances en Sistemas e Informática. Available from: <http://www.redalyc.org/articulo.oa?id=133114993004>

9. **Enrique Sucar Succar, Mario A. Romero Inzunza and Pilar Gómez-Gil.** Diseño de HEDEA: una herramienta para la construcción de Sistemas Tutores Inteligentes. Noviembre 2009 [online]. Available from: <https://ccc.inaoep.mx/~pgomez/publications/congress/MriInR09.pdf>
10. **Yaneidis Hernández Orozco.** Los Sistemas Tutores Inteligentes aplicados en la Educación. *Revista TINO ISSN 1995-9419* [online]. 30 May 2016. [Accessed 9 June 2017]. Available from: <https://revista.jovenclub.cu/los-sistemas-tutores-inteligentes-aplicados-en-la-educacion>
11. **Luisa Idorka Morales Rodríguez and Yarisleydi De Avila Fernández.** *Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos.* Universidad de las Ciencias Informáticas, 2012.
12. **Anita Ferreira, Pedro Salcedo.** La Arquitectura de ELE-TUTOR: Un Sistema Tutorial Inteligente para el Español como Lengua Extranjera. [online]. Available from: www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-09342012000200001
13. **María Lucía Barrón Estrada, Ramón Zatarain Cabada and Yasmín Hernández Pérez.** Tutor Inteligente con reconocimiento y manejo de emociones para Matemáticas *Revista Electrónica de Investigación Educativa.* . 2014. Vol. Vol. 16, no. Núm. 3.
14. **Ricardo Rosas, Rodrigo Zamorano, Miguel Nussbaum, Rodrigo Buzeta and Manuela Moulian, Jaime Bermeosolo.** Desarrollo y Evaluación de un Sistema Tutorial Inteligente para el Apoyo de la Enseñanza de la Lectura Inicial. [online]. Vol. Vol 6, no. 1. Available from: <http://www.psykhe.cl/index.php/psykhe/article/view/102>
15. **Lidiana Cruz Pazos, José Enrique Pacheco Piñeyro.** Sistema de enseñanza-aprendizaje inteligente de apoyo al curso introductorio de Matemática. Universidad de las Ciencias Informáticas, [no date].
16. **Laura Casas Fuentes, Olga Lidia Pérez González, Lisandra Docampo, Yailé Caballero Mota, Lenniet Coello, Isabel Yordi González and Angela**

Martín. SISTEMA INTELIGENTE PARA EL ALGEBRA LINEAL. Facultad de Informática, Univerisdad de Camagüey y Universidad APEC, [no date].

17. **Verónica Amela Tarongí.** Sistema Tutor Inteligente Adaptativo para Laboratorios Virtuales y Remotos [online]. Valencia : Universidad Politécnica de Valencia, Departamento de Ingeniería de Sistemas y Automática, 2010. Available from: <https://riunet.upv.es/bitstream/handle/10251/13851/Trabajo%20de%20Investigacion.pdf?sequence=1>
18. **José Manuel Gutiérrez.** Sistemas Expertos Basados en Reglas [online]. Dpto. de Matemática Aplicada. Universidad de Cantabria. Available from: <http://personales.unican.es/gutierjm/cursos/expertos/Reglas.pdf>
19. **Maykel Leyva.** Aplicación de la estrella Boehm turner a los proyectos de desarrollo de software en la Universidad de Guayaquil. [online]. Available from: <https://www.authorea.com/users/110721/articles/142954-aplicaci%C3%B3n-de-la-estrella-boehm-turner-a-los-proyectos-de-desarrollo-de-software-en-la-universidad-de-guayaquil>
20. **Ing. José Joskowicz.** Reglas y Prácticas en eXtreme Programming. [online]. Available from: <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
21. Qué es UML? | LENGUAJE DE MODELADO UNIFICADO UML. [online]. [Accessed 8 June 2017]. Available from: http://stadium.unad.edu.co/ovas/10596_9839/qu_es_uml.html
22. ¿Qué es PHP? [online]. Available from: <http://php.net/manual/es/intro-what-is.php>.
23. **Javier Eguíluz Pérez.** Introducción a javascript. [online]. Available from: www.librosweb.es/javascript.
24. MySQL | La base de datos de código abierto más popular | Oracle América Latina. [online]. [Accessed 9 June 2017]. Available from: <https://www.oracle.com/lad/mysql/index.html>

25. Introducción a Herramientas CASE Y System Architect. [online]. Available from: http://users.dsic.upv.es/asignaturas/eui/mtp/doc-practicas/intro_case_SA.pdf.
26. Visual Paradigm Para UML. [online]. Available from: <http://www.software.com.ar/p/visual-paradigm-para-uml#product-description>.
27. Apache HTTP Server: ¿Qué Es, Cómo Funciona Y Para Qué Sirve? [online]. 2014. Available from: <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.
28. ¿Qué Hace Un Servidor Web Como Apache? [online]. 2012. Available from: <http://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>.
29. **Dr. Daniel Gálvez Lio.** Curso de Sistemas Basados en el Conocimientos. Especializa en Inteligencias Artificial. Grupo de Investigación en Inteligencia Artificial Departamento de Ciencia de la Computación Facultad de Matemática, Física y Computación Universidad Central “Martha Abreu” de Las Villas. 1998.
30. Desarrollo de Software. Tarjetas CRC. [online]. 2012. Available from: <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>.
31. **Yunisleidy Fernández Romero.** Patrón modelo-Vista-Controlador.
32. **I. Sommerville.** Software Engineering.
33. **Patricia López, Francisco Ruiz.** Lenguaje Unificado de Modelado - UML. Universidad Cantabria – Facultad de Ciencias.
34. **Rolando Alfredo Hernández León y Sayda Coello González, 2011.** El proceso de investigación científica. -- Ciudad de La Habana : Editorial Universitaria.
35. **Ing. Carlos E. Lemus Serrano.** Estrategia metodológica para el desarrollo de un sistema tutor inteligente: módulo del dominio.

Anexos

Anexo 1. Historias de Usuario

Tabla 13. HU: Autenticar Usuario.

Historia de Usuario	
Nombre: Autenticar Usuario	
Número: 1	Usuario: Usuario
Prioridad: Alta	Puntos Estimados: 7
Descripción: El usuario se autentica en el sistema y se abre la sección correspondiente con los permisos según el rol que ocupe.	

Tabla 14. HU: Insertar Usuario.

Historia de Usuario	
Nombre: Insertar Usuario	
Número: 2	Usuario: Administrador
Prioridad: Alta	Puntos Estimados: 7
Descripción: El sistema debe buscar todos los estudiantes que coincidan con los datos especificados.	

Tabla 15. HU: Modificar Usuario.

Historia de Usuario	
Nombre: Modificar Usuario	
Número: 3	Usuario: Administrador
Prioridad: Alta	Puntos Estimados: 7
Descripción: El sistema debe permitir modificar los datos de los usuarios.	

Tabla 16. HU: Listar Usuario.

Historia de Usuario	
Nombre: Listar Usuario	
Número: 4	Usuario: Administrador
Prioridad: Alta	Puntos Estimados: 7
Descripción: El sistema debe listar todos los usuarios con sus datos.	

Tabla 17. HU: Eliminar Usuario.

Historia de Usuario	
Nombre: Eliminar Usuario	
Número: 5	Usuario: Administrador
Prioridad: Alta	Puntos Estimados: 7
Descripción: El sistema debe eliminar los datos de los usuarios.	

Tabla 18. HU: Insertar Estudiante.

Historia de Usuario	
Nombre: Insertar estudiante	
Número: 6	Usuario: Administrador
Prioridad: Alta	Puntos Estimados: 10
Descripción: El sistema debe insertar un nuevo estudiante (nombre, apellido, grupo, año académico).	

Tabla 19. HU: Listar Estudiantes.

Historia de Usuario	
Nombre: Listar estudiantes	
Número: 7	Usuario: Profesor
Prioridad: Baja	Puntos Estimados: 10
Descripción: El sistema debe listar todos los estudiantes.	

Tabla 20. HU: Buscar Datos del Estudiante.

Historia de Usuario	
Nombre: Buscar datos del estudiantes	
Número: 8	Usuario: Profesor
Prioridad: Baja	Puntos Estimados: 8
Descripción: El sistema debe buscar todos los estudiantes que coincidan con los datos especificados.	

Tabla 21. HU: Insertar Bibliografía.

Historia de Usuario	
Nombre: Insertar Bibliografía	
Número: 9	Usuario: Profesor y Administrador
Prioridad: Alta	Puntos Estimados: 14
Descripción: El sistema debe permitir almacenar toda la bibliografía a utilizar en la asignatura.	

Tabla 22. HU: Mostrar Bibliografía.

Historia de Usuario	
Nombre: Mostrar Bibliografía	
Número: 10	Usuario: Profesor y Administrador
Prioridad: Alta	Puntos Estimados: 7
Descripción: El sistema debe mostrarle al estudiante toda la bibliografía asignada una vez clasificado según el tipo de aprendizaje.	

Tabla 23. HU: Eliminar Bibliografía.

Historia de Usuario	
Nombre: Eliminar Bibliografía	
Número: 11	Usuario: Profesor y Administrador
Prioridad: Alta	Puntos Estimados: 7
Descripción: El sistema debe eliminar los datos de la bibliografía cuyo nombre coincida con el indicado (nombre).	

Tabla 24. HU: Insertar Cuestionario.

Historia de Usuario	
Nombre: Insertar Cuestionario	
Número: 12	Usuario: Profesor
Prioridad: Alta	Puntos Estimados: 10
Descripción: El sistema debe permitir insertar los cuestionarios para determinar el nivel de conocimientos de los estudiantes.	

Tabla 25. HU: Mostrar Cuestionario.

Historia de Usuario	
Nombre: Mostrar Cuestionario	
Número: 13	Usuario: Profesor
Prioridad: Alta	Puntos Estimados: 10
Descripción: El sistema debe mostrarle al estudiante el cuestionario a resolver.	

Tabla 26. HU: Eliminar Cuestionario.

Historia de Usuario	
Nombre: Eliminar Cuestionario	
Número: 14	Usuario: Profesor
Prioridad: Alta	Puntos Estimados: 8
Descripción: El sistema debe permitir eliminar el cuestionario cuyo nombre coincida con el indicado.	

Tabla 27. HU: Mostrar Protocolo Pedagógico.

Historia de Usuario	
Nombre: Mostrar Protocolo Pedagógico	
Número: 16	Usuario: Profesor
Prioridad: Alta	Puntos Estimados: 8
Descripción: El sistema debe mostrarle al estudiante toda la bibliografía asignada una vez clasificado según el tipo de aprendizaje.	

Tabla 28. HU: Eliminar Protocolo Pedagógico.

Historia de Usuario	
Nombre: Eliminar Protocolo Pedagógico	
Número: 17	Usuario: Profesor
Prioridad: Alta	Puntos Estimados: 8
Descripción: El sistema debe eliminar los datos de la bibliografía cuyo nombre coincida con el indicado (nombre).	

Tabla 29. HU: Modificar Protocolo Pedagógico.

Historia de Usuario	
Nombre: Modificar Protocolo Pedagógico	
Número: 18	Usuario: Profesor
Prioridad: Alta	Puntos Estimados: 8
Descripción: El sistema debe modificar los datos de la bibliografía a utilizar en la asignatura.	

Tabla 30. HU: Asignar Protocolo Pedagógico.

Historia de Usuario	
Nombre: Asignar Protocolo Pedagógico	
Número: 19	Usuario: Profesor
Prioridad: Alta	Puntos Estimados: 10
Descripción: El sistema asigna un protocolo pedagógico al estudiante según su estado cognoscitivo y su clasificación.	

Anexo 2. Tareas de Ingeniería

Tabla 31. Tarea de Ingeniería 1. Comprueba que el usuario y la contraseña existan en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 1	Número de HU: 1
Nombre de la Tarea: Comprueba que el usuario y la contraseña existan en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 02/01/2017	Fecha Fin: 04/01/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Comprueba que el usuario y la contraseña sean las correctas para permitir el acceso al sistema.	

Tabla 32. Tarea de Ingeniería 2. Le asigna el rol según los permisos que tenga el usuario.

Tarea de Ingeniería	
Número de Tarea: 2	Número de HU: 1
Nombre de la Tarea: Le asigna el rol según los permisos que tenga el usuario.	
Tipo de Tarea: Desarrollo	Estimación: 4 día
Fecha de Inicio: 05/01/2017	Fecha Fin: 08/01/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Dirige al usuario para la sección que le corresponda dependiendo del usuario y el rol que posea.	

Tabla 33. Tarea de Ingeniería 3. Comprobar que el usuario no esté creado en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 3	Número de HU: 2
Nombre de la Tarea: Comprobar que el usuario no esté creado en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 4 día
Fecha de Inicio: 09/01/2017	Fecha Fin: 12/01/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Comprueba que el usuario no esté registrado en el sistema.	

Tabla 34. Tarea de Ingeniería 4. Guardar los datos en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 4	Número de HU: 2
Nombre de la Tarea: Guardar los datos en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 13/01/2017	Fecha Fin: 15/01/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Guardar correctamente todos los datos del usuario en la base de datos.	

Tabla 35. Tarea de Ingeniería 5. Buscar el usuario en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 5	Número de HU: 3
Nombre de la Tarea: Buscar el usuario en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 4 día

Fecha de Inicio: 16/01/2017	Fecha Fin: 19/01/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Muestra todos los datos del usuario que cumplen con el criterio de búsqueda.	

Tabla 36. Tarea de Ingeniería 6. Modificar el rol que ocupa el usuario en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 6	Número de HU: 3
Nombre de la Tarea: Modificar el rol que ocupa el usuario en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 20/01/2017	Fecha Fin: 22/01/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: El administrador modifica el rol que ocupa el profesor en la base de datos.	

Tabla 37. Tarea de Ingeniería 7. Listar todos los usuarios del sistema.

Tarea de Ingeniería	
Número de Tarea: 7	Número de HU: 4
Nombre de la Tarea: Listar todos los usuarios del sistema.	
Tipo de Tarea: Desarrollo	Estimación: 7 día
Fecha de Inicio: 23/01/2017	Fecha Fin: 29/01/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Lista todos los usuarios que existen en el sistema con todos sus datos.	

Tabla 38. Tarea de Ingeniería 8. Buscar el usuario de la base de datos.

Tarea de Ingeniería	
Número de Tarea: 8	Número de HU: 5
Nombre de la Tarea: Buscar el usuario de la base de datos	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 30/01/2017	Fecha Fin: 01/02/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Busca el usuario en la base de datos.	

Tabla 39. Tarea de Ingeniería 9. Eliminar el usuario de la base de datos.

Tarea de Ingeniería	
Número de Tarea: 9	Número de HU: 5
Nombre de la Tarea: Eliminar el usuario de la base de datos	
Tipo de Tarea: Desarrollo	Estimación: 4 día
Fecha de Inicio: 02/02/2017	Fecha Fin: 05/02/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Eliminar todo los datos del usuario de la base de datos.	

Tabla 40. Tarea de Ingeniería 10. Comprobar que el estudiante no exista en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 10	Número de HU: 6
Nombre de la Tarea: Comprobar que el estudiante no exista en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 06/02/2017	Fecha Fin: 08/02/2017

Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo
Descripción: Verificar en la base de datos que no exista ningún estudiante con ese usuario.

Tabla 41. Tarea de Ingeniería 11. Guardar datos en la base de datos y mostrar sección correspondiente.

Tarea de Ingeniería	
Número de Tarea: 11	Número de HU: 6
Nombre de la Tarea: Guardar datos en la base de datos y mostrar sección correspondiente.	
Tipo de Tarea: Desarrollo	Estimación: 4 día
Fecha de Inicio: 09/02/2017	Fecha Fin: 12/02/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Guardar datos en la base de datos y mostrar la sección correspondiente al estudiante.	

Tabla 42. Tarea de Ingeniería 12. Salir del sistema en cualquier momento.

Tarea de Ingeniería	
Número de Tarea: 12	Número de HU: 6
Nombre de la Tarea: Salir del sistema en cualquier momento.	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 13/02/2017	Fecha Fin: 15/02/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Salir del sistema en cualquier momento.	

Tabla 43. Tarea de Ingeniería 13. Mostrar todos los datos de todos los estudiantes en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 13	Número de HU: 7
Nombre de la Tarea: Mostrar todos los datos de todos los estudiantes en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 10 día
Fecha de Inicio: 15/02/2017	Fecha Fin: 24/02/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Muestra los datos de todos los estudiantes que están registrados en el sistema.	

Tabla 44. Tarea de Ingeniería 14. Mostrar todos los estudiantes en la base de datos que cumplen con el criterio de búsqueda.

Tarea de Ingeniería	
Número de Tarea: 14	Número de HU: 8
Nombre de la Tarea: Mostrar todos los estudiantes en la base de datos que cumplen con el criterio de búsqueda.	
Tipo de Tarea: Desarrollo	Estimación: 8 día
Fecha de Inicio: 25/02/2017	Fecha Fin: 04/03/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Muestra todos los estudiantes que están registrados en el sistema y cumplen con el criterio de búsqueda.	

Tabla 45. Tarea de Ingeniería 15. Seleccionar la bibliografía para subir al sistema.

Tarea de Ingeniería

Número de Tarea: 15	Número de HU: 9
Nombre de la Tarea: Seleccionar la bibliografía para subirla al sistema.	
Tipo de Tarea: Desarrollo	Estimación: 7 día
Fecha de Inicio: 05/03/2017	Fecha Fin: 11/03/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Selecciona la bibliografía desde cualquier dirección y la agrega a la base de datos del sistema.	

Tabla 46. Tarea de Ingeniería 15. Rellenar los campos de clasificación de la bibliografía.

Tarea de Ingeniería	
Número de Tarea: 16	Número de HU: 9
Nombre de la Tarea: Rellenar los campos de clasificación de la bibliografía.	
Tipo de Tarea: Desarrollo	Estimación: 7 día
Fecha de Inicio: 12/03/2017	Fecha Fin: 18/03/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Rellenar los campos de clasificación de la bibliografía, para tener clasificada la bibliografía.	

Tabla 47. Tarea de Ingeniería 17. Listar bibliografía de la base de datos.

Tarea de Ingeniería	
Número de Tarea:17	Número de HU: 10
Nombre de la Tarea: Listar la bibliografía de la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 4 día
Fecha de Inicio: 18/03/2017	Fecha Fin: 21/03/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón	

Lauzardo
Descripción: Lista todos los documentos que existen en el sistema con todos sus clasificaciones.

Tabla 48. Tarea de Ingeniería 18. Muestra toda la bibliografía existente en el sistema.

Tarea de Ingeniería	
Número de Tarea:18	Número de HU: 10
Nombre de la Tarea: Muestra toda la bibliografía existente en el sistema.	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 22/03/2017	Fecha Fin: 24/03/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Muestra todos los documentos existentes en el sistema.	

Tabla 49. Tarea de Ingeniería 20. Buscar que el cuestionario no esté creado en la base de datos.

Tarea de Ingeniería	
Número de Tarea:20	Número de HU: 12
Nombre de la Tarea: Buscar que el cuestionario no esté creado en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 5 día
Fecha de Inicio: 01/04/2017	Fecha Fin: 05/04/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Buscar que no exista un cuestionario creado con el mismo identificador en el sistema.	

Tabla 50. Tarea de Ingeniería 21. Crear el nuevo cuestionario en la base de datos.

Tarea de Ingeniería

Número de Tarea: 21	Número de HU: 12
Nombre de la Tarea: Crear el nuevo cuestionario en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 5 día
Fecha de Inicio: 06/04/2017	Fecha Fin: 10/04/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Crea un nuevo cuestionario.	

Tabla 51. Tarea de Ingeniería 22. Listar todos los cuestionarios existentes en la base de datos.

Tarea de Ingeniería	
Número de Tarea:22	Número de HU: 13
Nombre de la Tarea: Listar todos los cuestionarios existentes en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 10 día
Fecha de Inicio: 11/04/2017	Fecha Fin: 20/04/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Muestra todos los cuestionarios que existen en el sistema.	

Tabla 52. Tarea de Ingeniería 23. Eliminar el cuestionario específico existente en la base de datos.

Tarea de Ingeniería	
Número de Tarea:23	Número de HU: 14
Nombre de la Tarea: Eliminar el cuestionario específico existente en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 8 día
Fecha de Inicio: 21/04/2017	Fecha Fin: 28/04/2017

Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo
Descripción: Elimina el cuestionario seleccionado.

Tabla 53. Tarea de Ingeniería 24. Listar bibliografía de la base de datos.

Tarea de Ingeniería	
Número de Tarea: 24	Número de HU: 15
Nombre de la Tarea: Listar la bibliografía de la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 29/04/2017	Fecha Fin: 01/05/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Lista todos los documentos que existen en el sistema con todos sus clasificaciones.	

Tabla 54. Tarea de Ingeniería 25. Insertar la bibliografía necesaria para ese protocolo pedagógico.

Tarea de Ingeniería	
Número de Tarea: 25	Número de HU: 15
Nombre de la Tarea: Insertar la bibliografía necesaria para ese protocolo pedagógico.	
Tipo de Tarea: Desarrollo	Estimación: 3 día
Fecha de Inicio: 02/05/2017	Fecha Fin: 04/05/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Se selecciona toda la bibliografía necesaria.	

Tabla 55. Tarea de Ingeniería 26. Buscar que el protocolo pedagógico no este creado.

Tarea de Ingeniería	
Número de Tarea: 26	Número de HU: 15
Nombre de la Tarea: Buscar que el protocolo pedagógico no este creado.	
Tipo de Tarea: Desarrollo	Estimación: 2 día
Fecha de Inicio: 05/05/2017	Fecha Fin: 06/05/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Buscar que el nuevo protocolo pedagógico que se creará no exista en el sistema.	

Tabla 56. Tarea de Ingeniería 27. Listar todos los protocolos pedagógicos existentes en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 27	Número de HU: 16
Nombre de la Tarea: Lista todos los protocolos pedagógicos existentes en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 8 día
Fecha de Inicio: 07/05/2017	Fecha Fin: 14/05/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Muestra una lista con todos los protocolos pedagógicos que existen en el sistema.	

Tabla 57. Tarea de Ingeniería 28. Eliminar protocolo pedagógico específico existente en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 28	Número de HU: 17

Nombre de la Tarea: Eliminar protocolo pedagógico específico existente en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 8 día
Fecha de Inicio: 15/05/2017	Fecha Fin: 22/05/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Eliminar protocolo pedagógico específico del sistema.	

Tabla 58. Tarea de Ingeniería 29. Comprobar que el protocolo pedagógico esté creado en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 29	Número de HU: 18
Nombre de la Tarea: Comprobar que el protocolo pedagógico esté creado en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 4 día
Fecha de Inicio: 22/05/2017	Fecha Fin: 25/05/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Buscar que el protocolo pedagógico exista en la base de datos.	

Tabla 59. Tarea de Ingeniería 30. Modificar los datos en la base de datos.

Tarea de Ingeniería	
Número de Tarea: 30	Número de HU: 18
Nombre de la Tarea: Modificar los datos en la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 4 día
Fecha de Inicio: 25/05/2017	Fecha Fin: 29/05/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón	

Lauzardo
Descripción: Modifica los datos del protocolo pedagógico seleccionado.

Tabla 60. Tarea de Ingeniería 31. Asigna el protocolo pedagógico correspondiente al estudiante.

Tarea de Ingeniería	
Número de Tarea: 31	Número de HU: 19
Nombre de la Tarea: Asigna el protocolo pedagógico correspondiente al estudiante.	
Tipo de Tarea: Desarrollo	Estimación: 5 día
Fecha de Inicio: 30/05/2017	Fecha Fin: 03/06/2017
Programador Responsable: Lourdes A. Risco Ferrer y Luis E. Manchón Lauzardo	
Descripción: Asigna el protocolo pedagógico que le corresponde al estudiante según su clasificación y su estado cognoscitivo.	

Anexo 3. Ejemplos de Casos de pruebas de aceptación

Tabla 75. HU16_P1. Mostrar Protocolo Pedagógico

Caso de prueba de aceptación	
Código: HU16_P1	Historia de Usuario: 16
Nombre: Mostrar Protocolo Pedagógico	
Descripción: Prueba para la funcionalidad Mostrar Protocolo Pedagógico	
Condiciones de Ejecución: El protocolo pedagógico debe estar creado en el sistema.	
Entrada/ Pasos de Ejecución: El sistema muestra una lista con todos los protocolos pedagógicos existentes en el sistema.	
Resultado Esperado: El protocolo pedagógico muestra toda la bibliografía por la cual está formado.	
Evaluación de la Prueba: Prueba Satisfactoria	

Tabla 76. HU17_P1. Eliminar Protocolo Pedagógico.

Caso de prueba de aceptación	
Código: HU17_P1	Historia de Usuario: 17
Nombre: Eliminar Protocolo Pedagógico	
Descripción: Prueba para la funcionalidad Eliminar Protocolo Pedagógico	
Condiciones de Ejecución: El protocolo pedagógico debe estar creado en el sistema.	
Entrada/ Pasos de Ejecución: Cuando se muestren todos los protocolos pedagógicos se selecciona el que se desea eliminar.	
Resultado Esperado: El protocolo pedagógico es eliminado de la base de datos.	
Evaluación de la Prueba: Prueba Satisfactoria	

Tabla 77. HU18_P1. Modificar Protocolo Pedagógico.

Caso de prueba de aceptación	
Código: HU18_P1	Historia de Usuario: 18
Nombre: Modificar Protocolo Pedagógico	
Descripción: Prueba para la funcionalidad Modificar Protocolo Pedagógico	
Condiciones de Ejecución: -	
Entrada/ Pasos de Ejecución: Se selecciona el protocolo pedagógico y se edita la bibliografía, se aumenta o se elimina bibliografía.	
Resultado Esperado: El protocolo pedagógico es modificado de la base de datos.	
Evaluación de la Prueba: Prueba Satisfactoria	

Tabla 78. HU19_P1. Asignar Protocolo Pedagógico.

Caso de prueba de aceptación	
Código: HU19_P1	Historia de Usuario: 19
Nombre: Asignar Protocolo Pedagógico	
Descripción: Prueba para la funcionalidad Asignar Protocolo Pedagógico	
Condiciones de Ejecución: Deben estar creadas las reglas en el sistema para asignarle a los estudiantes el protocolo pedagógico	
Entrada/ Pasos de Ejecución: Se le muestra al estudiante el protocolo pedagógico que debe utilizar, según la clasificación individual y el estado cognoscitivo que posee.	
Resultado Esperado: El protocolo pedagógico es modificado de la base de datos.	
Evaluación de la Prueba: Prueba Satisfactoria	