

Universidad de las Ciencias Informáticas



Facultad 2

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

“Sistema tutorial inteligente para el apoyo de la enseñanza de la asignatura de Teleinformática”

Autora: Geicy Acosta Guzmán

Tutores: Ing. Víctor Alejandro Roque Domínguez

Ing. Lourdes Alicia Risco Ferrer

La Habana, mayo 2019

“Año 61 de la Revolución”

Lo esencial es invisible a los ojos...

Antoine de Saint-Exupéry

Agradecimientos

Quiero agradecer a mis padres, hermano y familia por ser los mejores del mundo y apoyarme en toda esta etapa tan importante de mi vida.

Gracias tesoro por insistirme que estudie cuando deseos no tenía.

Agradezco a mis suegros por estar siempre pendientes como si fuera su hija.

Con todo mi corazón le agradezco a mis tutores por dedicarme su tiempo, paciencia y apoyo, y por ayudarme cada vez que lo necesité, no tengo como agradecerles.

Al tribunal y oponente por compartir su sabiduría y experiencia al corregirme errores.

A los profesores que han compartido conmigo las aulas de esta hermosa universidad creada por nuestro comandante Fidel Castro, a él gracias.

A mis amigos Devorat, Lisardo y el Pepe por ser incondicionales.

A todos ustedes quiero dar mis más profundos agradecimientos.

Dedicatoria

Por amarme, cuidarme, y enseñarme mucho de la vida, quiero dedicar este trabajo de todo corazón a mis padres Mercedes Guzmán y Jorge Acosta.

A mi hermano Joyce Acosta por ser mi ejemplo de perseverancia y el regalo más importante que me han dado mis padres.

A mis abuelas María y Tina, por ser mi hito en la construcción de una hermosa familia.

A mis tías que han sido como mis madres desde que nací y a mis primos por mantenernos juntos como hermanos.

A mis suegros Lily y Adónis por ser maravillosos conmigo y apoyarme en todos estos años de estudio.

A mi gordo por compartir su vida conmigo.

Resumen

El desarrollo de las Tecnologías de las Informáticas y las Comunicaciones (TICs), ha tenido gran influencia en las diversas esferas sociales, destacándose su uso en la educación, incorporando elementos novedosos que han revolucionado los antiguos métodos de enseñanza-aprendizaje.

Actualmente la Universidad de las Ciencias Informáticas (UCI), se encuentra inmersa en la búsqueda de nuevas formas que permitan facilitar el proceso de formación de los estudiantes, teniendo en cuenta el estado cognitivo, estilos de aprendizajes, tipos de inteligencias y conocimientos previos de los mismos, para proporcionar la ayuda pedagógica adecuada.

El objetivo de este trabajo, comprende el desarrollo de un Sistema Tutorial Inteligente (STI) para la asignatura Teleinformática. 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 y la interfaz de usuario. Se utilizó para su desarrollo como técnica de Inteligencia Artificial (IA) los Sistemas Basados en Reglas.

Para el desarrollo de este sistema se realizó un análisis preliminar de los conceptos más importantes asociados a los sistemas tratados con el fin de entenderlos mejor; se siguió la metodología XP y se utilizaron herramientas y tecnologías como el entorno de desarrollo integrado NetBeans, como lenguaje de programación php, como sistema gestor de base de datos MySQL y como servidor web Apache

Palabras Claves: Herramientas educativas, Sistemas basados en reglas, Sistema Tutorial Inteligente, Teleinformática.

Índice

Introducción	1
Capítulo 1. Fundamentación teórica	5
1.1 Herramientas educativas	5
1.2 Sistemas Tutoriales Inteligentes	5
1.3 Sistemas tutoriales inteligentes en la educación	9
1.4 Técnica de la Inteligencia Artificial	13
1.5 Tecnologías a utilizar	14
1.5.1 Metodología de desarrollo de software.....	14
1.5.2 Lenguaje de modelado.....	14
1.5.3 Herramienta de modelado.....	15
1.5.4 Lenguaje de programación.....	15
1.5.5 Framework.....	15
1.5.6 Herramienta de administración de Base de Datos	16
1.5.7 Sistema Gestor de Base de Datos	16
1.5.8 Servidor web	16
1.5.9 Entorno de Desarrollo Integrado	17
Conclusiones parciales.....	17
Capítulo 2. Propuesta de solución.	18
2.1 Propuesta de solución.	18
2.2. Diseño del Sistema Basado en Reglas	23
2.3. Usuarios relacionados con el sistema.....	25
2.4. Fase de exploración.....	26
2.4.1. Historias de usuario	26
2.4.2 Requisitos no funcionales del sistema.....	29
2.5. Fase de planificación	30

2.5.1. Plan de iteraciones.....	30
Conclusiones Parciales:.....	31
Capítulo 3: Diseño, implementación y prueba.	32
3.1 Fase de diseño	32
3.1.1 Tarjetas CRC	32
3.1.2 Estilo arquitectónico	34
3.1.3 Patrones de diseño	36
3.2 Modelo físico de la Base de Datos.....	37
3.3 Despliegue del sistema.....	39
3.4 Tareas de ingeniería.....	39
3.5 Pruebas	43
3.5.1 Pruebas de rendimiento	43
3.5.2 Pruebas de aceptación	45
Conclusiones Parciales.....	47
Conclusiones Generales.....	48
Recomendaciones	49
Referencias Bibliográficas.....	50

Índice de Tablas

Tabla 1: Descripción de los Estilos de Aprendizaje. Fuente: (26).....	21
Tabla 2 Instrumento para la determinación de estilos de aprendizaje propuesta. Fuente (26)	22
Tabla 3: Usuarios relacionados en el sistema.....	25
Tabla 4: Insertar protocolo pedagógico.....	27
Tabla 5: Mostrar protocolo pedagógico.....	28
Tabla 6: Eliminar protocolo pedagógico.....	28
Tabla 7: Modificar protocolo pedagógico.....	28
Tabla 8: Asignar protocolo pedagógico.....	29
Tabla 9: Plan de Iteraciones.....	30
Tabla 10: Tarjeta CRC: Gestionar usuario.....	33
Tabla 11: Tarjeta CRC: Gestionar estudiante.....	33
Tabla 12: Tarjeta CRC: Gestionar bibliografía.....	33
Tabla 13: Tarjeta CRC: Gestionar cuestionario.....	34
Tabla 14: Tarjeta CRC: Gestionar protocolo.....	34
Tabla 15: Tarea de Ingeniería 19. Eliminar la bibliografía de la base de datos.....	43
Tabla 16: Resultados de las pruebas de rendimiento.....	44
Tabla 17: Caso de prueba de aceptación HU14_P1. Insertar protocolo pedagógico.....	45
Tabla 18: Resultado de las pruebas.....	46

Índice de Figuras

Figura 1: Relación entre los módulos de un STI. Fuente: (6).	6
Figura 2: Módulo del tutor. Fuente: (6)	7
Figura 3: módulo del estudiante. Fuente: (6).....	8
Figura 4: módulo del dominio. Fuente: (6).....	9
Figura 5 Técnicas de IA utilizadas en los STI estudiados. Fuente: Elaboración propia.	12
Figura 6: Propuesta de solución del Sistema. Fuente: Elaboración propia.	18
Figura 7: Arquitectura de un sistema basado en reglas. Fuente: Elaboración propia.	23
Figura 8: Diseño de la arquitectura MVC. Fuente: Elaboración propia.	36
Figura 9: Modelo físico de la base de datos.	38
Figura 10: Diagrama de Despliegue.....	39
Figura 11: Pruebas de aceptación	47

Introducción

La transición de un modelo educativo propio de una sociedad industrializada a un modelo educativo marcado por las demandas de una sociedad informatizada, es un proceso que está viviendo la mayoría de las instituciones a nivel mundial(1). El diseño e implementación de programas de capacitación docente que utilicen las Tecnologías de la Información y las Comunicaciones (TIC), efectivamente son un elemento clave para lograr reformas educativas profundas y de amplio alcance. Las instituciones de formación docente deberán optar entre asumir un papel de liderazgo en la transformación de la educación, o bien quedar atrás en el continuo cambio tecnológico.

De igual manera las TIC tienen una serie de ventajas para el alumnado evidentes como: la posibilidad de interacción que ofrecen, por lo que se pasa de una actitud pasiva por parte del alumnado a una actividad constante, a una búsqueda y replanteamiento continuo de contenidos y procedimientos, también aumentan la implicación del alumnado en sus tareas y desarrollan su iniciativa, ya que se ven obligados constantemente a tomar "pequeñas" decisiones, a filtrar información, a escoger y seleccionar.

Es importante destacar que el uso de las TIC favorece el trabajo colaborativo con los iguales, el trabajo en grupo, no solamente por el hecho de tener que compartir ordenador con un compañero o compañera, sino por la necesidad de contar con los demás en la consecución exitosa de las tareas encomendadas por el profesorado (2).

En las universidades, desde hace tiempo también se ha tratado de aplicar lenguajes y técnicas de programación propia de la Inteligencia Artificial (IA) para el diseño de sistemas expertos en educación o Sistemas Tutoriales Inteligentes (STI), y se han llegado a desarrollar muchos de estos sistemas en conexión con los avances de la tecnología hipermedia y multimedia. En los últimos tiempos los avances en el desarrollo de la telemática, la inteligencia artificial y la informática educativa están convergiendo hacia el diseño de sistemas hipermedia adaptativos y tutores-asistentes para entornos virtuales de enseñanza (3).

La Universidad de las Ciencias Informáticas (UCI) posee un alto grado de tecnología destinada a la labor docente-educativa, que sin dudas sirve de apoyo para el desarrollo de herramienta educativas que apoyen y mejoren el proceso de enseñanza-aprendizaje. Las herramientas educativas son programas diseñados con el propósito de apoyar la labor de los profesores en el proceso de enseñanza docente educativo. Dentro del plan de estudio de la carrera Ingeniería en Ciencias Informáticas, se encuentra como asignatura lectiva del 3er año Teleinformática. Entre los objetivos de esta asignatura se encuentran:

- Caracterizar la arquitectura TCP/IP y el funcionamiento de sus protocolos para optimizar el desempeño de los servicios telemáticos.

- Seleccionar los recursos técnicos necesarios para la instalación de una red de área local, basándose en normas internacionales considerando el costo económico.
- Aplicar las tecnologías, estándares y protocolos de redes a diversos casos de estudio basándose en las características funcionales y estructurales de dichas tecnologías, donde demuestren la utilización práctica de las mismas en un ambiente cercano a la realidad laboral.

La plataforma Zera actualmente es la que se utiliza como herramienta educativa que apoya al proceso de enseñanza aprendizaje, sin embargo, para la interacción con la misma se necesita de la guía constante del profesor. Además, cuando los estudiantes realizan el autoestudio lo hacen por su cuenta y no tienen la presencia del profesor para que oriente este proceso, por lo que deben consultar la bibliografía y materiales de apoyo. Estos, por lo general, no conocen cuál es la bibliografía ajustada a sus características y necesidades para consultar, por lo que presentan dificultades a la hora de retener el contenido. Todo lo antes mencionado repercute directamente en el rendimiento académico reflejado en las notas que obtienen en la asignatura y el desarrollo de sus habilidades.

Teniendo en cuenta la situación problemática anterior, se plantea el siguiente **problema a resolver**: Las insuficiencias de los recursos educativos en la asignatura Teleinformática dificulta el aprendizaje de los estudiantes del 3er año de la Facultad 2 de la UCI.

El **objeto de estudio** lo constituye: las herramientas educativas; quedando definido como **campo de acción**: los sistemas tutoriales inteligentes en la asignatura de Teleinformática.

De ahí, que se define como **objetivo general de la investigación**: desarrollar un sistema tutorial inteligente como apoyo a la enseñanza en la asignatura de Teleinformática en la Facultad 2 de la Universidad de las Ciencias Informáticas.

Como estrategia para cumplir el objetivo planteado se proponen las siguientes **tareas de investigación**:

1. Definición de los referentes teóricos metodológicos que sustentan el proceso de enseñanza aprendizaje apoyados en los Sistemas Tutoriales Inteligentes.
2. Caracterización de los diferentes componentes que integran un Sistema Tutorial Inteligente.
3. Identificación de las diferentes técnicas de Inteligencia Artificial, así como las tecnologías y herramientas para el desarrollo del Sistema Tutorial Inteligente.
4. Implementación del Sistema Tutorial Inteligente según la metodología XP.

5. Constatación en la práctica de la contribución y funcionamiento del STI para el apoyo de la enseñanza de la asignatura de Teleinformática en la Facultad 2 en la Universidad de las Ciencias Informáticas.

Con el objetivo de alcanzar la solución deseada y de facilitar el desarrollo de la investigación se combinaron diferentes **métodos de investigación**.

Métodos teóricos

Analítico-Sintético: este método es usado para realizar un análisis de libros, sitios web, documentos electrónicos y otras fuentes bibliográficas que relacionan elementos relevantes para la investigación, como las principales herramientas educativas y las características y conceptos más importantes de los Sistemas Tutoriales Inteligentes asociados a la educación.

Histórico-Lógico: Este método es ha empleado en la investigación para consultar la bibliografía referente a los STI, la trayectoria, evolución y comportamiento en la educación. Condujo a estructurar la documentación investigada de una manera organizada, permitiendo un mejor entendimiento de la información consultada en la misma.

Modelación: Facilitó la creación de modelos, representando de manera gráfica parte del contenido de la presente investigación.

Métodos empíricos

Entrevista: Se realizaron entrevistas a los profesores de la asignatura Teleinformática para conocer las necesidades del sistema.

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

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 Anexos, donde se abarca todo lo relacionado con la investigación realizada.

Capítulo 1. Fundamentación Teórica: Este capítulo está dedicado a los principales conceptos asociados a la problemática, se realiza un análisis de la metodología de desarrollo de software, se estudian las diferentes formas en que los Sistemas Tutoriales Inteligentes son utilizados como herramientas educativas y se examinan los principales métodos, técnicas y herramientas asociadas al tema en cuestión.

Capítulo 2. Propuesta de solución: 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 1. Fundamentación teórica

En el presente capítulo se aborda un grupo de conceptos de vital importancia para el desarrollo de la investigación, destacándose las herramientas educativas, principalmente los Sistemas Tutores Inteligentes. Se exponen los resultados del análisis de la arquitectura de los STI investigados, así como las técnicas de IA que aplican. Se define la técnica de IA a utilizar para el sistema a desarrollar. Se describen las herramientas y metodología que serán utilizadas para resolver el problema antes expuesto, con el objetivo de obtener información que permita establecer el marco de trabajo favorable para el desarrollo de una solución aplicable al problema científico planteado.

1.1 Herramientas educativas

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 ventajas se encuentran:

- Fomento de la comunicación profesor/alumno: La relación profesor/alumno, restringida tradicionalmente 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 la sensación de tener un canal de comunicación con el alumno permanentemente abierto.
- Facilidades para el acceso a la información y la entrega de actividades: En la medida en que cualquier información relacionada con la asignatura está disponible en la web, el alumno puede acceder a la misma en cualquier momento y desde cualquier lugar. Puede así el alumno, efectuar los envíos en cualquier momento, mientras que en los sistemas tradicionales los tiempos de entrega quedan limitados al momento de clase o al horario de tutorías presenciales.
- Fomento del debate y la discusión: El hecho de extender la docencia más allá del aula, así como la ductilidad de las aplicaciones que la plataforma proporciona permiten fomentar la participación de los alumnos a la hora de analizar problemáticas jurídicas de interés (4).

1.2 Sistemas Tutoriales Inteligentes

Los Sistemas Tutoriales Inteligentes (STI) se definen como un sistema de enseñanza asistida por computadora, que utiliza técnicas de Inteligencia Artificial, principalmente para representar el conocimiento y dirigir una estrategia de enseñanza. El objetivo de un STI es que mediante el uso de técnicas de

Inteligencia Artificial se le dé la capacidad a una computadora de involucrarse en el proceso de enseñanza, convirtiéndola en un instructor dinámico capaz de simular a un profesor humano (5).

La educación a distancia no se quedó atrás en el uso de esta nueva tecnología y se integró a internet como un medio para mejorar la calidad educativa y el perfeccionamiento a distancia, donde convergen un gran número de estudiantes de diferentes edades y de todos los perfiles socio-culturales. Debido a esto la arquitectura de los STI debe soportar el procesamiento distribuido, para poder utilizar la red de manera eficiente.

Una mejora sustancial en el uso de las plataformas para educación a distancia existe debido a la provisión de información relevante y el mejoramiento de la experiencia educativa, a través de nuevos recursos pedagógicos que facilita el proceso cognitivo del estudiante. A esto, se le puede sumar un entorno flexible con acceso a la red desde cualquier punto donde el STI se encuentre almacenado. Esto también puede significar una reducción de costos importantes (en materiales y principalmente en tiempo (6).

Diversos autores plantean una arquitectura de los sistemas tratados que se constituye de 3 módulos, el módulo del experto o dominio, el módulo del estudiante y el módulo de instrucción o tutor, cuenta además con una interface de usuario.

La estructura general de la interacción de los módulos de un sistema tutorial inteligente es:

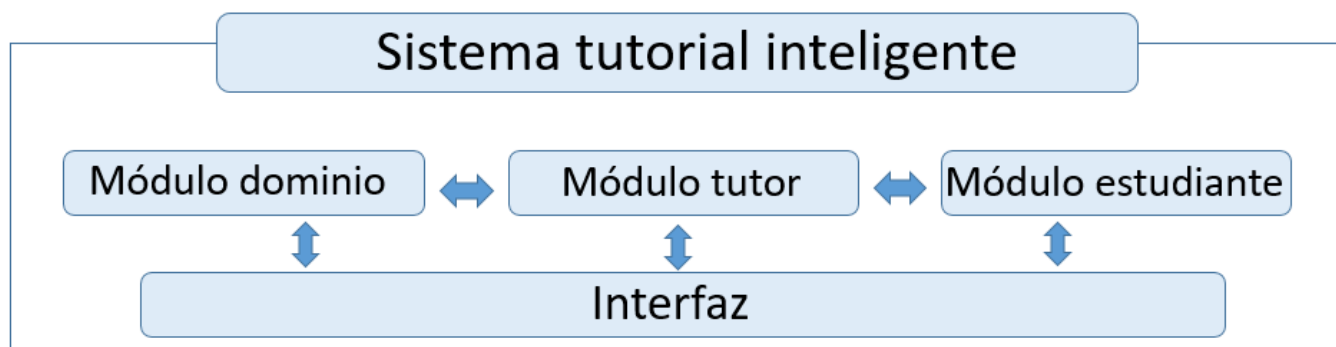


Figura 1: Relación entre los módulos de un STI. Fuente: (6).

En un STI el **módulo tutor** define y aplica una estrategia pedagógica de enseñanza, contiene los objetivos a ser alcanzados y los planes utilizados para alcanzarlos. Selecciona los problemas, monitorea el desempeño, provee asistencia y selecciona el material de aprendizaje para el estudiante. Integra el conocimiento acerca del método de enseñanza, las técnicas didácticas y del dominio a ser enseñado. Consta de:

- Protocolos pedagógicos: almacenados en una base de datos, con un gestor para la misma.
- Planificador de lección: organiza los contenidos de la misma.
- Analizador de perfil: analiza las características del alumno, seleccionando la estrategia pedagógica más conveniente.

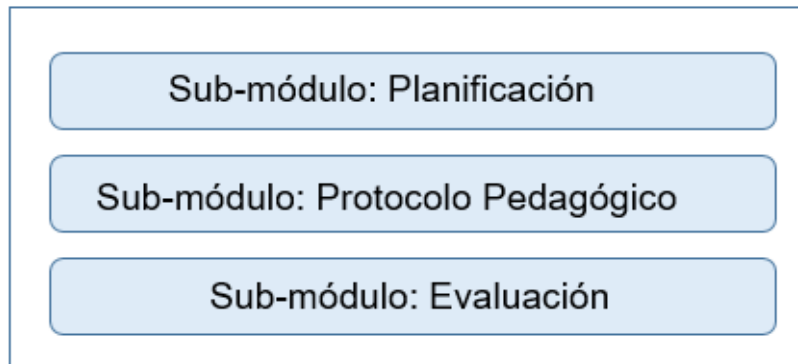


Figura 2: Módulo del tutor. Fuente: (6)

El **módulo estudiante** del STI tiene por objetivo para una adecuada retroalimentación del sistema, la realización del modelado y del diagnóstico cognitivo del alumno. Se ha planteado para el módulo estudiante los siguientes submódulos (ambos almacenan los datos en una base de datos del estudiante, con un gestor para la misma).

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 y que progresivamente el actualizador de conocimientos irá modificando a través de los resultados obtenidos en las evaluaciones efectuadas por el módulo del tutor quien le enviará dichos resultados procesados.

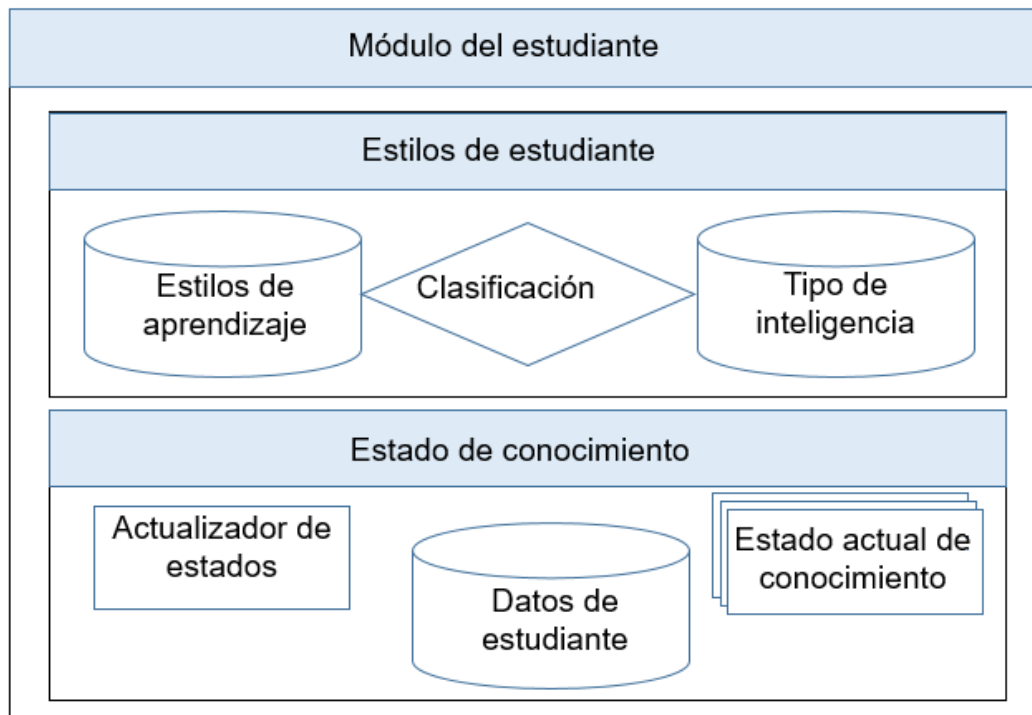


Figura 3: módulo del estudiante. Fuente: (6)

El **módulo dominio** tiene el objetivo global de almacenar todos los conocimientos dependientes e independientes del campo de aplicación del STI.

Puede tener los submódulos 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: material multimedia que se requiere para facilitarle al alumno apropiarse de conocimiento en la sesión pedagógica, dígase las imágenes, videos, sonidos.

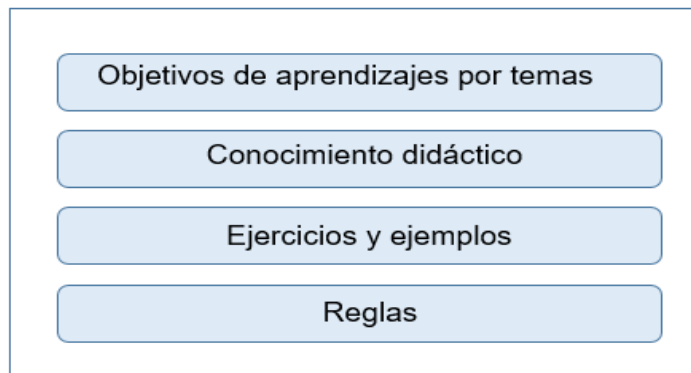


Figura 4: módulo del dominio. Fuente: (6)

1.3 Sistemas tutoriales inteligentes en la educación

Lo siguiente son sistemas tutoriales inteligentes existentes a nivel internacional y nacional, analizados de acuerdo a su objetivo, su estructura y las técnicas de la inteligencia artificial que utilizaron.

- **Sistema tutorial inteligente para la asignatura Reconocimiento de Patrones.**

Herramienta de apoyo al proceso de enseñanza-aprendizaje de la asignatura optativa Reconocimiento de Patrones, la cual ayuda a consolidar los contenidos del plan de estudio definido para la asignatura. Desarrollado en la Universidad de las Ciencias Informáticas y fue estructurado en cuatro módulos: módulo del estudiante (está investigación recibe ya el estudiante clasificado, por lo que este módulo, solo realiza el diagnóstico cognoscitivo del estudiante), módulo del dominio, módulo pedagógico y la interfaz. Este STI para asignarle al estudiante el material de estudio utiliza el Sistema Basado en Reglas como técnica de IA (7).

- **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 de un Sistema Instruccionales Inteligente (ITS: *Intelligent Tutoring System.*),

modelo del dominio, modelo del alumno y modelo pedagógico. Se utilizó como técnica de Inteligencia Artificial un Sistema Basado en Reglas para regular el flujo de actividades asociadas a cada niño (8).

- **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 tres módulos, módulo tutor, módulo estudiante, módulo de dominio y la interfaz; como técnica de Inteligencia Artificial utiliza Redes Bayesianas y Sistemas Basados en Reglas (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 como técnica de Inteligencia Artificial utiliza Razonamiento Basado en Casos (RBC).

- **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, así como los cinco problemas tipos del Álgebra Lineal. 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 (10).

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

Es un Sistema Tutorial 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 Tutorial 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 tutorial 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 (11).

- **Sistema Tutorial 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 módulo del alumno (evalúa el conocimiento y la actualización del alumno en el sistema), el módulo del tutor (selecciona la intervención educativa más adecuada al alumno) y el módulo 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 (12).

- **STI-SO**

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. Desarrollado en la Universidad de las Ciencias Informáticas y fue estructurado en tres módulos: módulo del estudiante, módulo del dominio, módulo pedagógico y una interfaz. Este STI para asignarle al estudiante el material de estudio utiliza el RBC como técnica de IA (13).

- **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 de la muestra de ciento ochenta y dos (182) estudiantes. El software 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 (14).

Luego de una breve selección y análisis de los sistemas tutoriales inteligentes anteriores utilizados para la enseñanza en la educación, se puede afirmar que la mayoría de los STI, generalmente están compuestos por tres módulos: módulo tutor, módulo estudiante, módulo dominio y además cuenta con una interfaz de usuario.

Como se evidencia en el gráfico de la figura 5, existe igualdad en la cantidad de sistemas analizados que utilizan la técnica de IA razonamiento basado en casos y Sistemas basados en reglas. El razonamiento basado en casos es un enfoque que aborda nuevos problemas tomando como referencia problemas similares resueltos en el pasado. De modo que problemas similares tienen soluciones similares, y la similitud juega un rol esencial. Sus componentes fundamentales son la base de casos, el módulo de recuperación de casos y el módulo de adaptación de las soluciones (15). No se utilizó la técnica porque no se cuenta con una base de datos histórica; siendo un componente fundamental para la aplicación de esta técnica en del diseño de sistemas inteligente. Por lo que se utilizó el Sistema Basado en Reglas (SBR), ya que sí se 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.

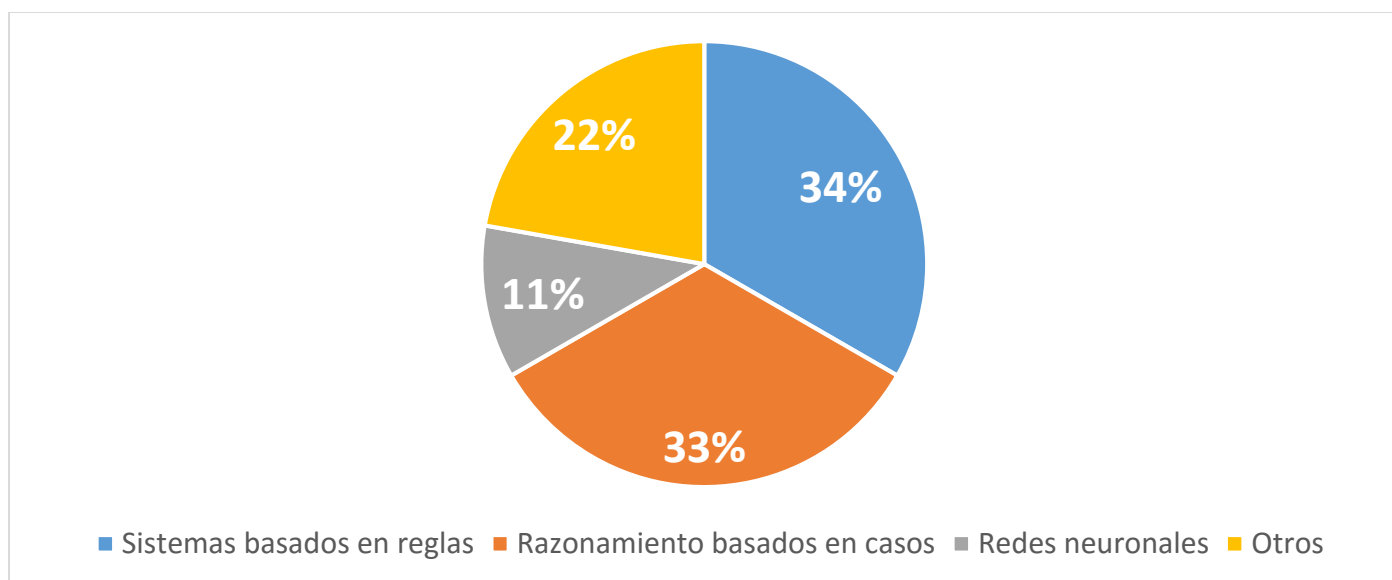


Figura 5 Técnicas de IA utilizadas en los STI estudiados. Fuente: Elaboración propia.

1.4 Técnica de la 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.**

Los SBR son Sistemas Basados en el Conocimiento (SBC) en los que la forma de representación del conocimiento usado son las reglas de producción y como método de inferencia utiliza la regla de Modus Ponens.

Las reglas utilizan un formato IF - THEN para representar el conocimiento, la parte IF de una regla es una condición (también llamada premisa o antecedente), y la parte THEN de la regla (también llamada acción, conclusión o consecuente) permite inferir un conjunto de hechos nuevos si se verifican las condiciones establecidas en la parte IF (16).

Las reglas pueden expresar un amplio rango de asociaciones:

- Situación – acción
- Premisa – conclusión
- Antecedente – consecuente

Según (17) los SBR son frecuentemente confundidos con sistemas lógicos; sin embargo, estos se definen en dos ideas principales:

- Los SBR son generalmente no monotónicos, es decir los hechos pueden variar su veracidad durante el proceso de razonamiento.
- Los SBR aceptan incertidumbre en el proceso deductivo.

En este tipo de sistema, la Base de Conocimiento (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 (17).

1.5 Tecnologías a utilizar

Para la realización del proyecto, fue necesario la utilización de varias herramientas y tecnologías. Las cuales se mencionan y describen brevemente.

1.5.1 Metodología de desarrollo de software

Se empleará 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 (18).

1.5.2 Lenguaje de modelado

El Lenguaje Unificado de Modelado cuenta con una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Es un lenguaje gráfico, que tiene como fin especificar y documentar un sistema de software de un modo estándar, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema (19).

1.5.3 Herramienta de modelado

Visual Paradigm v15.1

Hoy en día existen herramientas CASE que soportan UML como lenguaje de modelado. Su utilización en el desarrollo de software aumenta en gran medida la productividad y reduce costos en cuanto a tiempo y dinero. Visual Paradigm es una de estas herramientas, usada para el desarrollo de software y gestión de empresas, que ofrece todas las características que demanda la arquitectura empresarial, la gestión de proyectos, el desarrollo de software y la colaboración en equipo en una solución única. Es multiplataforma ya que está disponible tanto para Windows como para Linux, cabe mencionar que cuenta con una versión pagada y una open source (20). Esta herramienta fue utilizada para el modelado del diagrama de despliegue del STI y para el modelado de la base de datos.

1.5.4 Lenguaje de programación

PHP v5.5.26

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 (21).

1.5.5 Framework

Bootstrap v3.3.7

Bootstrap es un marco para ayudar a diseñar sitios web de forma más rápida y sencilla. Incluye plantillas de diseño basadas en HTML y CSS para tipografía, formularios, botones, tablas, navegación, modales, carruseles de imágenes, etc. También le brinda soporte para los complementos de JavaScript. Fue utilizado para el diseño de las vistas.

1.5.6 Herramienta de administración de Base de Datos

PhPMyAdmin v4.8.5

PhPMyAdmin es una herramienta de software gratuita, escrita en PHP, diseñada para manejar la administración de MySQL a través de la Web. PhpMyAdmin admite una amplia gama de operaciones en MySQL y MariaDB. Las operaciones de uso frecuente (gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc.) se pueden realizar a través de la interfaz de usuario, mientras que todavía tiene la capacidad de ejecutar directamente cualquier instrucción SQL (22). Fue utilizada para la gestión de la base de datos (crearla y editarla).

1.5.7 Sistema Gestor de Base de Datos

MySQL v5.7

MySQL es un SGBD cliente-servidor, que usa programación multihilo y permite la conexión de múltiples usuarios de manera concurrente. Se compone de un servidor SQL, varios programas clientes y bibliotecas, herramientas administrativas, y una gran variedad de interfaces de programación. Se puede obtener también como una biblioteca multihilo que se enlaza dentro de otras aplicaciones para obtener un producto más pequeño, más rápido, y más fácil de manejar (23).

Ventajas:

- Dispone de borrados multitablas.
- Mejores utilidades de administración (salvas y recuperación de errores), contando con un sistema de replicación multihilo en los servidores esclavos.
- Los registros de replicación contienen los marcadores de transacción necesarios para asegurarse que las transacciones son replicadas apropiadamente.
- Recuperación automática ante fallas.

1.5.8 Servidor web

Apache v2.4

En lo que concierne a las tecnologías por parte del servidor sobresale Apache programa que administra servidores para alojar sitios Web. Según el fabricante, alrededor del cincuenta por ciento de los sitios de internet está almacenado en servidores Apache. Es estable, seguro y eficiente, software libre y de código

abierto para plataformas Unix, Windows y otras, e implementa el protocolo HTTP. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Apache es un servidor de red para el protocolo HTTP, elegido para poder funcionar como un proceso independiente, sin que eso solicite el apoyo de otras aplicaciones o directamente del usuario. Para poder hacer esto, una vez que se haya iniciado, crea unos subprocesos (que normalmente vienen llamados "children processes") para poder gestionar las solicitudes: estos, sin embargo, no podrán nunca interferir con el proceso mayor. Trabaja con Perl, PHP, Java y páginas. jsp, teniendo todo el soporte que se necesita para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto (24).

1.5.9 Entorno de Desarrollo Integrado

NetBeans v8.2

NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. *Sun Microsystems* fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos. NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso (25).

Conclusiones parciales

Como resultado del análisis bibliográfico y la investigación realizada a los distintos STI, tanto internacionales como nacionales, y 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 Teleinformática.
- 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 2. Propuesta de solución.

El presente capítulo tiene como objetivo reflejar las actividades que se realizaron en el proceso de análisis y diseño de la propuesta de solución. Se encuentran entre los elementos a destacar los módulos que componen el sistema y como se utilizan. En el mismo se exponen los artefactos más importantes que describen el funcionamiento del sistema, tales como las historias de usuarios que rigen el desarrollo de la solución, detallando la información del análisis y del diseño de la solución en cuestión.

2.1 Propuesta de solución.

Se propone como solución el desarrollo de un STI de apoyo a la asignatura de Teleinformática, el 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 6.

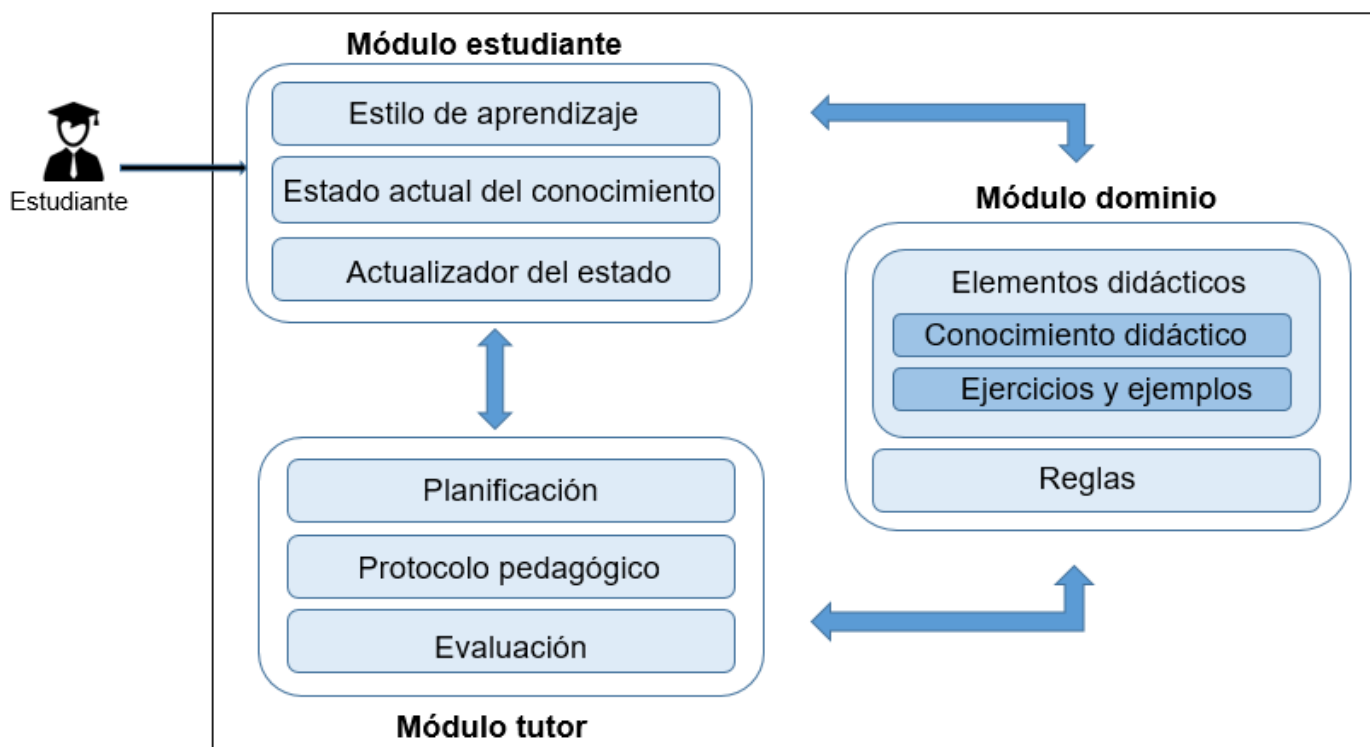


Figura 6: Propuesta de solución del Sistema. Fuente: Elaboración propia.

Visto en la figura anterior, el estudiante al ser autenticado realiza una encuesta para asignarle el estilo de aprendizaje, luego realiza un cuestionario para conocer el dominio que presenta sobre la temática de la asignatura que debe vencer. Las evaluaciones de los diferentes cuestionarios fueron propuestas por los

profesores de la asignatura de Teleinformática, las cuales están divididas en bien (B), regular (R) y mal (M), según el por ciento de las respuestas correctas 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.

Dependiendo de la evaluación obtenida en el cuestionario para conocer el dominio de la asignatura, 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. Tomando como antecedente: el estilo de aprendizaje y la evaluación obtenida en el cuestionario de dominio de la asignatura, y el consecuente: el protocolo pedagógico correspondiente a esos objetivos.

Ejemplo: Si estudiante pertenece a Clase1 (Estilo de aprendizaje) y Evaluación (X) entonces asigna Protocolo Y.

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 submódulos: 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 cuestionario. 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 cuestionario 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 cuestionario 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: cuando el estudiante se autentica en el STI realiza un test para clasificarlo según su estilo de aprendizaje, en correspondencia con el resultado del cuestionario; el sistema utiliza esa clasificación para asignarle a los estudiantes sus respectivos protocolos pedagógicos.

El test utilizado para identificar los estilos de aprendizaje fue el desarrollado por la doctora Zulma Cataldi (26) para estudiantes de ingeniería, este es una versión del propuesto por Felder y Silverman. Además, Zulma Cataldi propone de los diez estilos de aprendizaje, la utilización de ocho. En la tabla 1, se listan los estilos de aprendizaje y la descripción de estos.

Tabla 1: Descripción de los Estilos de Aprendizaje. Fuente: (26)

Estilos de Aprendizaje	Descripción
Activo/Reflexivo	Los estudiantes con estilo de aprendizaje activo, prefieren el desarrollo de ejercicios que leer y les gusta el trabajo en grupos; en cambio los reflexivos trabajan en grupo y prefiere la teoría. Aprenden mejor observando y reflexionando.
Sensitivo/Intuitivo	Los estudiantes con estilo de aprendizaje sensitivo les agradan aprender hechos y se les facilita solucionar problemas por métodos bien establecidos y no les gustan las complicaciones o sorpresas, a los estudiantes intuitivos no les agrada mucho los cursos que implican memorización y cálculos de rutina.
Secuencial/Global	Los estudiantes con estilo de aprendizaje secuencial aprenden mejor con pequeños pasos incrementales, mientras que los estudiantes con estilo de aprendizaje global requiere de una visión integral.
Visual/Verbal	Los estudiantes con estilo de aprendizaje visual prefieren imágenes, gráficos y representaciones visuales por lo que recuerda. Mientras que el estudiante con estilo de aprendizaje verbal prefieren explicaciones escritas o habladas

Las respuestas 1-4 darán la característica del estilo activo-reflexivo. Específicamente las preguntas 2 y 3 medirán el estilo activo y las preguntas 1 y 4 el reflexivo. Considerando el estilo activo y al reflexivo, al igual que los demás, como dos extremos de un continuo. Las respuestas 5-8 darán las características del estilo sensitivo-intuitivo. Específicamente las preguntas 5 y 6 miden el estilo sensitivo; las preguntas 7 y 8 el estilo

intuitivo. Las respuestas 9-12 darán las características del estilo visual-verbal. Específicamente las 9 y 11 son para el estilo verbal y las 10 y 12 para el estilo visual. Las preguntas 13-16 darán la característica global-secuencial. En particular las preguntas 13 y 15 se refieren al estilo global en tanto que las 14 y 16 secuencial. Como este cuestionario se basa fundamentalmente en la naturaleza excluyente que tienen los sucesos correspondientes a cada par de estilos, para resolver en qué lugar de cada continuo se encuentra un alumno se deben sumar los puntajes de las respuestas con estilos homólogos y promediarlos respecto de 8.

Diseño del cuestionario

Tabla 2 Instrumento para la determinación de estilos de aprendizaje propuesta. Fuente (26)

Preguntas	Puntaje				
	1 Nunca, 2 Raramente, 3 Ocasionalmente, 4 Usualmente, 5 Siempre				
	1	2	3	4	5
1. Cuando resuelvo un problema leo y releo el enunciado hasta descifrar su solución, sin escribir nada.					
2. Cuando resuelvo un problema, enseguida escribo los datos, las incógnitas, y las fórmulas que conozco.					
3. Me expreso y participo en clase siempre que es posible.					
4. En general, escucho y pienso acerca de lo que otros dicen en clase, pero rara vez participo.					
5. Las lecciones que relaten hechos históricos o actuales me resultan fáciles de aprender.					
6. Si debo hacer un cálculo, desisto en seguida cuando veo que no puedo resolverlo.					
7. Al hacer un cálculo que no es sencillo, trato de resolverlo a toda costa.					
8. Cuando en clase otros exponen soluciones de problemas o expresan sus opiniones puedo, con cierta facilidad, corregir sus errores o detectar que no son lógicas.					
9. En un examen, suelo recordar como una fotografía el texto escrito que estudié.					
10. Muchas veces logro entender un tema cuando puedo hacer un dibujo o diagrama con él.					
11. Lo que habla el profesor de un tema, es muy importante para mí, pues es como mejor entiendo.					

12. Cuando el profesor explica, lo que más recuerdo es lo que dibujó en el pizarrón.					
13. A veces cuando resuelvo un problema, no puedo explicar cómo lo hice.					
14. Cuando se explica un tema nuevo, hasta que el profesor no lo resume agrupándolo en varios títulos seguidos, me resulta confuso.					
15. A veces, cuando se explica un tema nuevo, me surgen preguntas sobre cosas que todavía faltan explicar o que se relacionan con temas de otras materias.					
16. Cuando resuelvo un problema, escribo cada paso en forma ordenada y prolija para entenderlo mejor.					

- 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.

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 7.

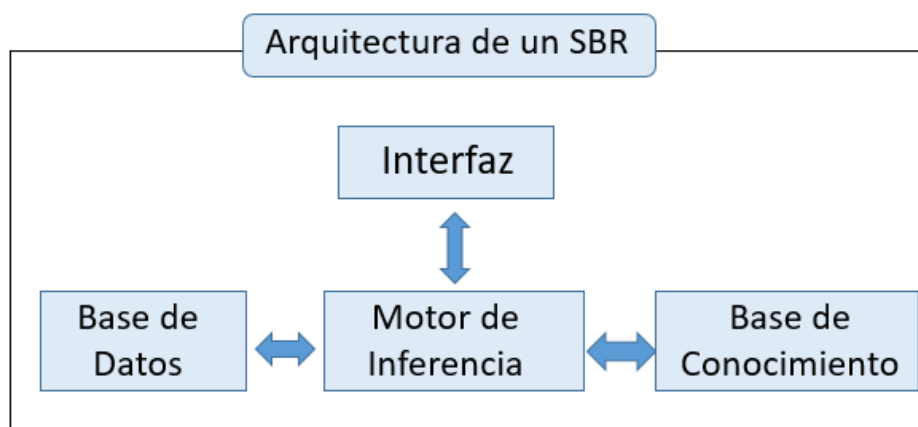


Figura 7: Arquitectura de un sistema basado en reglas. Fuente: Elaboración propia.

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 coteja 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.

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

En la Tabla 3 se muestra, una descripción general que realizan los diferentes usuarios en el sistema.

Tabla 3: Usuarios relacionados en 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

La fase de exploración es la fase en la que se define el alcance del proyecto, además de realizar la entrevista con los clientes, definiéndose las llamadas Historias de Usuarios (HU) y con ellas la determinación de los requisitos planteados por el cliente, así como la que el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Esta es una fase en la cual su duración oscila de pocas semanas a pocos meses, dependiendo del conocimiento que tenga el equipo desarrollador.

2.4.1. Historias de usuario

Entre los artefactos que define la metodología seleccionada se encuentran las Historias de Usuario. Se utilizan para especificar los requisitos de las aplicaciones software en las metodologías ágiles (SCRUM, XP, FDD, ASD, AUP, LD, etc.). Las HU son tarjetas en donde el interesado describe brevemente (con el fin de que sean dinámicas y flexibles) las características que el sistema debe poseer, sean requisitos funcionales o no funcionales (27).

Las HU quedan estructuradas de la siguiente manera:

- **Nombre:** nombre descriptivo de la HU.
- **Número:** número de la HU.
- **Usuario:** nombre de usuarios que interacciona.
- **Prioridad en el Negocio:** grado de prioridad que le asigna el cliente a la HU en dependencia de sus necesidades. Puede tomar los valores de: alta, media o baja.
 - 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**: unidades de tiempo estimadas por el equipo de desarrollo para darle cumplimiento a la HU (en días).
- **Descripción**: descripción simple que brinda el cliente sobre lo que debe hacer la funcionalidad en cuestión.

A continuación, se muestra el nombre de cada una de las historias de usuario del sistema y la descripción de algunas de las HU más importantes. El resto se puede apreciar en el [Anexo 1](#).

- 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.
- HU 19: Asignar protocolo pedagógico

Tabla 4: 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.	

Tabla 5: 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 6: 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 7: 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 8: 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.	

2.4.2 Requisitos no funcionales del sistema

Los Requisitos no funcionales se refieren a los requisitos relacionados con la calidad del software, especifican propiedades del sistema, como restricciones del entorno o la implementación, rendimiento, dependencia de plataformas, facilidad de mantenimiento, extensibilidad y fiabilidad (28).

Software

- Se requiere que la aplicación sea multiplataforma.
- **Requerimientos para el servidor:** Es necesario la instalación del gestor de base de datos MySQL v5.7, lenguaje de programación PHP v7.0 y la herramienta de administración de base de datos PhpMyAdmin v4.8.3.

Hardware para el Servidor:

La computadora 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 9: 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 Mostrar protocolo pedagógico	6 semanas

	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.

Capítulo 3: Diseño, implementación y prueba.

Introducción

El diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. El objetivo del diseño es producir un modelo o representación de una entidad que se va a construir posteriormente. Una vez concluida la fase de análisis y diseño de la propuesta de solución se procede a la implementación de las clases y ejecución de casos de prueba que evalúen las funcionalidades de la herramienta de configuración. En el capítulo se visualiza como se llevó a cabo la implementación del sistema, definiéndose diferentes tareas por iteraciones.

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 mejor solución para el diseño de un software es la que se basa en cumplir con las expectativas del cliente de manera exitosa, presentando la menor cantidad de clases y métodos posibles garantizando su usabilidad. Siguiendo los principios de la metodología escogida se utiliza la técnica para el modelamiento de clases representada por la Tarjetas CRC, las cuales permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. El uso de esta técnica permite guiar el diseño de la aplicación a través de las clases que se examinan y filtran en base a sus responsabilidades para con el sistema, así como las clases con las que se relacionan para completar dichas responsabilidades (29).

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 algunas de las tarjetas CRC:

Tabla 10: Tarjeta CRC: Gestionar usuario

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

Tabla 11: Tarjeta CRC: Gestionar estudiante

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

Tabla 12: Tarjeta CRC: Gestionar bibliografía

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

Tabla 13: Tarjeta CRC: Gestionar cuestionario

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

Tabla 14: Tarjeta CRC: Gestionar protocolo

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

3.1.2 Estilo arquitectónico

El diseño arquitectónico define la relación entre los elementos estructurales principales del software, los patrones de diseño que se pueden utilizar para lograr los requisitos que se han definido para el sistema, y las restricciones que afectan a la manera en que se pueden aplicar los patrones de diseño arquitectónicos, mientras que la arquitectura de software describe la estructura del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos (30).

Atendiendo a las propiedades generales de la aplicación y los requerimientos no funcionales se decidió seleccionar el patrón Modelo-Vista-Controlador (MVC) el cual separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Modelo:

Representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos actuando como “intermediario” con la base de datos (30).

EL modelo fue dividido en dos:

Datos: es donde se encuentran las clases encargadas de almacenar la información en la base de datos.

- **conexion**: clase encargada de establecer la conexión con la base de datos.
- **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.
- **Bibliografía**: clase encargada de almacenar toda la bibliografía.

Negocio: es donde se encuentran las clases encargadas de registrar la información en la base de datos.

- **Usuario**: clase encargada de registrar los usuarios.
- **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.
- **Protocolo_est_bibliografía**: 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:

Es la encargada de mostrar la información al usuario de forma gráfica y “humanamente legible”.

Controlador:

Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario, de forma “humanamente legible” (30).

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

- **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.

En la figura 8 se representa el funcionamiento de la arquitectura MVC, donde el usuario le hace la solicitud al controlador, este invoca al modelo para consultar la base de datos, el modelo le envía al controlador lo solicitado para que procese y envíe a la vista la información, ya que esta es la encargada de entregársela al cliente de una forma entendible.

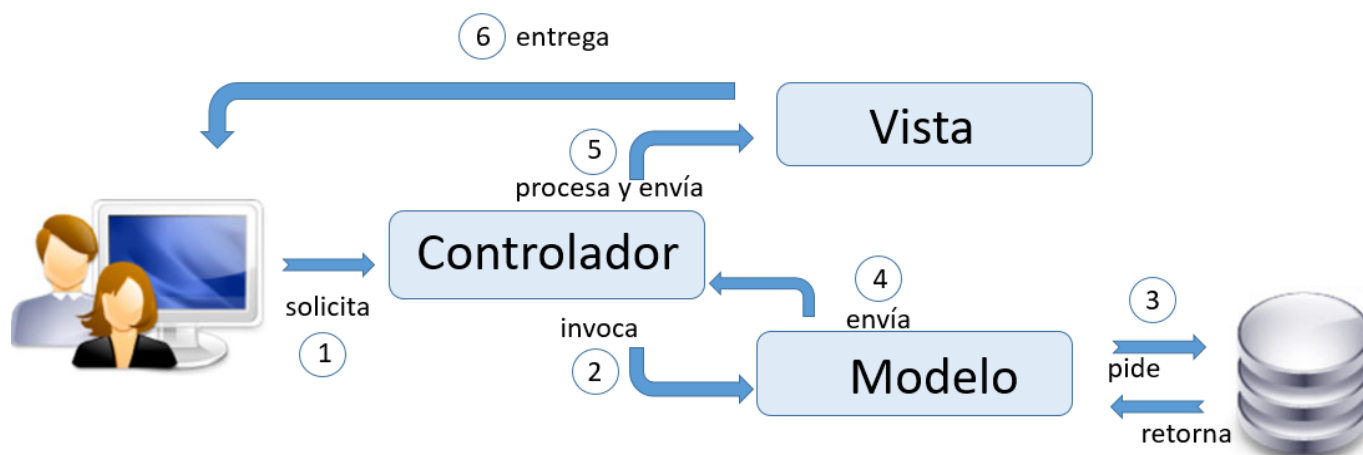


Figura 8: Diseño de la arquitectura MVC. Fuente: Elaboración propia.

3.1.3 Patrones de diseño

Los patrones GRASP codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos expresados en forma de patrones (31).

Los Patrones de Asignación de Responsabilidades (GRASP) constituyen una colección de principios de diseño orientados a objetos que guían la asignación de responsabilidades sobre los objetos, tienen como objetivo esencial ayudar a entender el diseño de éstos, aplicando el razonamiento de diseño de una manera metódica, racional y explicable (31).

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 delega en el método crea_estudiante() de la clase estudiante e insert_profesor() de la clase profesor.

El patrón **Bajo Acoplamiento** es el encargado de asignar una responsabilidad para mantener bajo acoplamiento con el objetivo de tener las clases lo menos ligadas entre sí, de forma que, si se realiza modificación en alguna de ellas, tenga la mínima repercusión en el resto de clases. Esto se evidencia en todas las clases, ya que un cambio en las clases modelo no implica cambios en las controladoras ni en las vistas.

El patrón **Alta Cohesión** es el encargado de asignar una responsabilidad de modo que la cohesión siga siendo alta, con el objetivo de que cada una de las clases tenga agrupada todas las funciones similares en clases individuales. Ejemplo de esto tenemos la clase controladora conexion la cual realiza todas las operaciones de conexión con la base de datos.

3.2 Modelo físico de la Base de Datos

El diseño de la base de datos es fundamental para el desarrollo de cualquier aplicación. A continuación, se muestra el modelo físico de la base de datos para el STI.

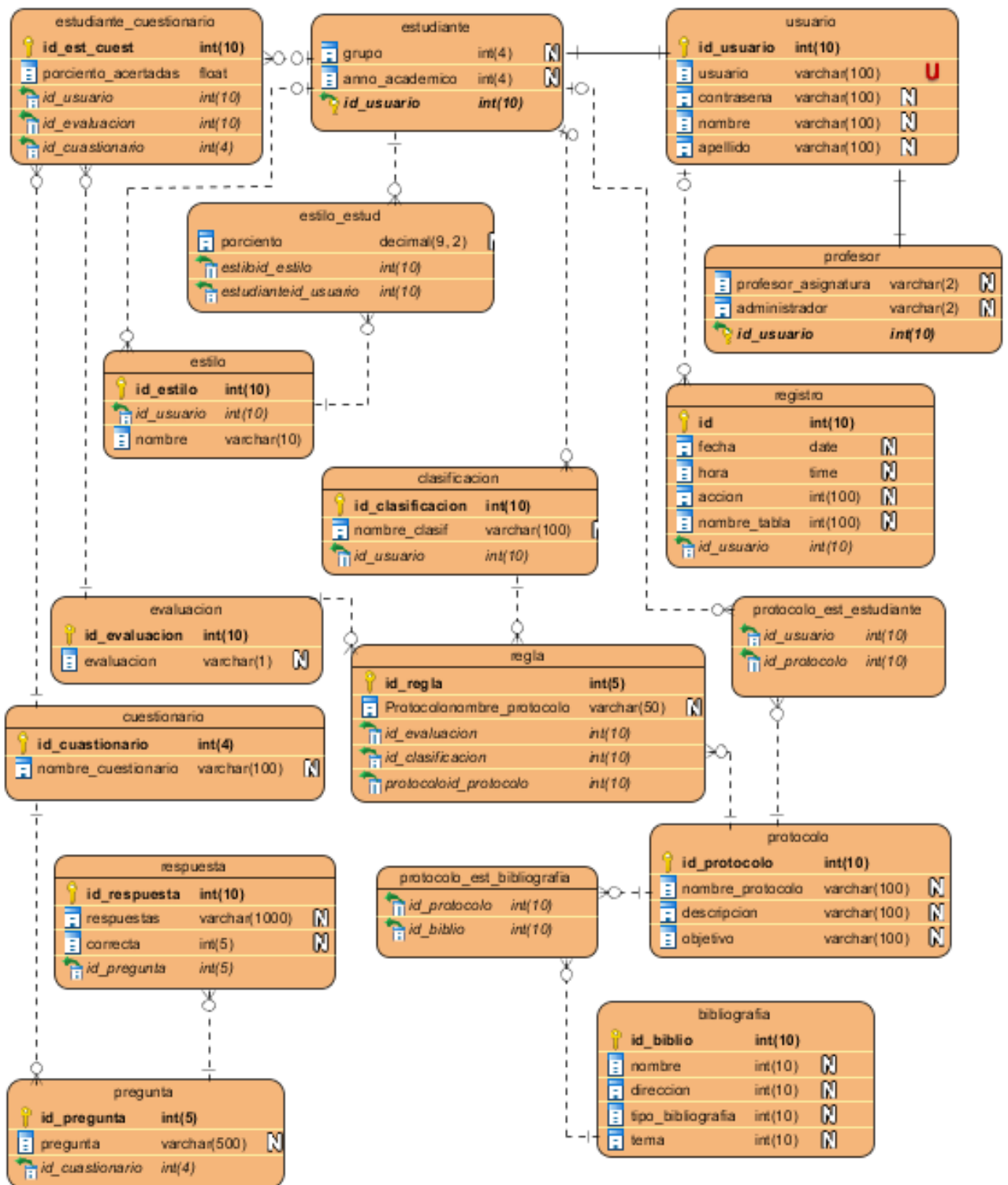


Figura 9: Modelo físico de la base de datos.

3.3 Despliegue del sistema

El modelo de despliegue es parte de los diagramas que ayuda a comprender la arquitectura de un sistema. Su propósito es el de distribuir el sistema, asignando componentes ejecutables a nodos en el diagrama de despliegue. Es la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de hardware sobre el que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituye el sistema físico (32).

A continuación, se presenta el diagrama de despliegue del STI. Está representado por tres componentes principales, el cliente, el servidor web y la una base de datos. El cliente se conecta al servidor mediante el puerto 80 y este a su vez se conecta mediante el protocolo TCP con la base de datos.

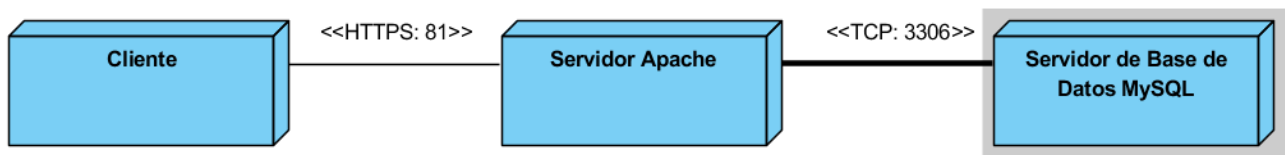


Figura 10: Diagrama de Despliegue.

3.4 Tareas de ingeniería

Se describieron 32 tareas de Ingeniería que permitieron representar un nivel descriptivo más profundo de las Historias de usuario. Están asociadas independientemente de las Historias de Usuario y definieron los detalles esenciales que las componen (33).

El empleo de este artefacto permitió agilizar los procesos de diseño e implementación, debido a que en estas se detallaron los siguientes elementos:

- **Número de tarea:** Se especifica un valor numérico que representa el identificador de la tarea.
- **Número de la Historia de usuario:** Representa la HU asociada a la tarea.
- **Nombre de tarea:** Define el nombre de la tarea.
- **Tipo de tarea:** Las tareas de ingeniería pueden ser de tipo: Desarrollo, Corrección, Mejora u otra especificación definida según las necesidades.

- **Puntos estimados:** Estos valores están dados según la estimación de tiempo realizada por el desarrollador; teniendo en cuenta la complejidad de la tarea y que cada punto representa el transcurso de un día.
- **Fecha inicio:** Fecha en la que se comienza el desarrollo de la tarea.
- **Fecha fin:** Fecha en la que se culmina el desarrollo de la tarea.
- **Programador responsable:** Se especifica el nombre del programador encargado de su desarrollo.
- **Descripción:** Se describe formalmente los elementos que debe cumplir la tarea.

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 9. Insertar bibliografía, 10. Mostrar bibliografía y 11. Eliminar bibliografía.

HU 9. 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 10. 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 11. 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 12. Insertar cuestionario, 13. Mostrar cuestionario y 14. Eliminar cuestionario.

HU 12. 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 13. Mostrar cuestionario, se desarrollan las siguientes tareas:

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

HU 14. 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 15. Insertar protocolo, 16. Mostrar protocolo, 17. Eliminar protocolo, 18. Modificar protocolo y 19. Asignar protocolo.

HU 15. 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 16. Mostrar protocolo, se desarrollan las siguientes tareas:

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

HU 17. Eliminar protocolo, se desarrollan las siguientes tareas:

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

HU 18. 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 19. 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 19 Eliminar la bibliografía de la base de datos, el resto de las descripciones se encuentran en el [Anexo 2](#).

Tabla 15: Tarea de Ingeniería 19. Eliminar la bibliografía de la base de datos.

Tarea de Ingeniería	
Número de la Tarea: 19	Número de la HU: 11
Nombre de la Tarea: Eliminar la bibliografía de la base de datos.	
Tipo de Tarea: Desarrollo	Estimación: 7 día
Fecha Inicio: 25/3/2019	Fecha Fin: 31/3/2019
Programador responsable: Geicy Acosta Guzmán	
Descripción: Eliminar la bibliografía de la base de datos.	

3.5 Pruebas

Los sistemas de software hoy en día son parte importante e integral en la gran mayoría de las actividades diarias, es por ello que se debe tener en cuenta que los sistemas o aplicaciones son creadas, desarrolladas e implementadas por seres humanos y, por ende, en cualquiera de sus etapas de creación se puede presentar una equivocación que puede llevar a defectos en las aplicaciones. Las pruebas son necesarias porque con ellas se puede ayudar a reducir los riesgos en las aplicaciones y lograr de esta manera que se identifiquen los defectos antes de que se ejecuten (34).

3.5.1 Pruebas de rendimiento

En la Ingeniería del Software, las pruebas de rendimiento son aquellas que son realizadas para determinar qué tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas de rendimiento constituyen un subconjunto de la Ingeniería de Pruebas, la cual se esfuerza en mejorar el rendimiento, basándose en el diseño y la arquitectura de un sistema, antes de la realización del proceso de codificación (35).

3.5.1.1 Diseño de pruebas de rendimiento

Las pruebas de rendimiento pueden servir para diferentes propósitos:

- Demostrar que el sistema cumple los criterios de rendimiento.
- Comparar dos sistemas para encontrar cuál de ellos funciona mejor.
- Medir que partes del sistema o de carga de trabajo provocan que el conjunto rinda mal.

Para su diagnóstico, los ingenieros de software utilizan herramientas como pueden ser monitorizaciones que midan qué partes de un dispositivo o software contribuyen más al mal rendimiento o para establecer niveles del mismo que mantenga un tiempo de respuesta aceptable.

Para realizar una valoración de rendimiento para el STI, es necesario definir un entorno de prueba. Se tiene una computadora con un procesador Intel(R) Core(TM) i7-4510U, CPU a 2.00 GHz - 2.60 GHz de velocidad y 16 GB de memoria RAM. La computadora tiene instalado el sistema operativo Windows 10. Para esto se generaron registros de eventos para todo el proceso del sistema, con el objetivo de ver el tiempo de generación de dichos registros de eventos y llegar a la siguiente conclusión.

Para el desarrollo de este tipo de prueba se utilizó la herramienta Jmeter ya que es empleada para rendimientos en el servidor web Apache. Se configuró la herramienta para hacer pruebas con 25, 50 y 100 conexiones concurrentes, los resultados se muestran en la tabla 16 (36).

Tabla 16: Resultados de las pruebas de rendimiento

Cantidad de hilos	Número de muestra	Media de tiempo en milisegundos para cada una de las muestras	Tiempo total para atender los hilos (s)	Tiempo promedio para cada hilo (s)	Rendimiento medio en Kbyte por segundo
100	2040	924	1885	19	239.81
50	2040	300	612	12	239.14
25	2040	30	319	2	319.1

Luego de ejecutadas las pruebas de rendimiento se observa que el sistema responde adecuadamente ante la concurrencia de hasta 100 conexiones. El rendimiento, en cuanto al tiempo total para atender los hilos disminuye al tener más conexiones, pero no en un factor considerable. Por lo que se consideran los resultados dentro de los parámetros.

3.5.2 Pruebas de aceptación

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 el [Anexo 3](#).

Tabla 17: Caso de prueba de aceptación HU14_P1. Insertar protocolo pedagógico

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 este 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.

3.5.2.1 Resultado de las pruebas de aceptación

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 (NC) por iteración:

Tabla 18: Resultado de las pruebas.

Iteración	NC destacadas	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 15 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.

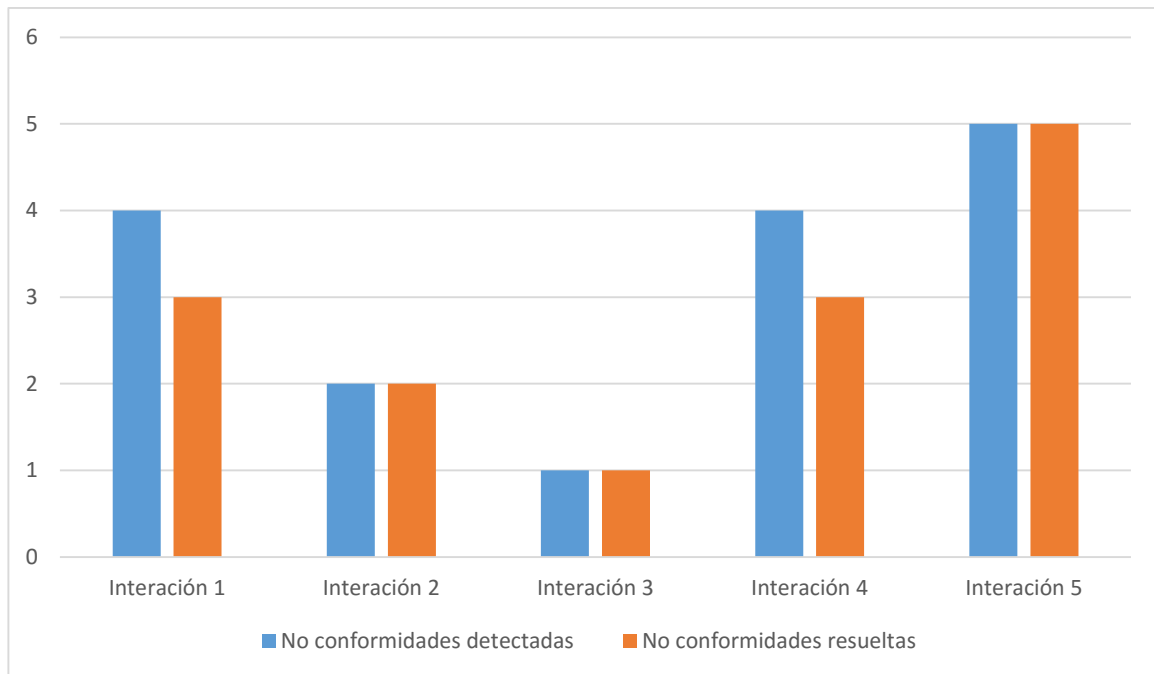


Figura 11: Pruebas de aceptación

Conclusiones Parciales

En este capítulo se abordó la etapa de diseño, implementación y prueba del Sistema Tutorial Inteligente para la asignatura de Teleinformática, 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 de aceptación permitieron verificar que el sistema realiza correctamente los requisitos solicitados por el cliente.

Conclusiones Generales

Al finalizar esta investigación se llega a la conclusión de que se le dio solución al problema de investigación cumpliendo con el objetivo propuesto al principio de la misma, pues:

- Se desarrolló una aplicación web, capaz de ayudar a los estudiantes de la asignatura de Teleinformática a consolidar su conocimiento y lograr aprovechar su tiempo de auto estudio, mostrándoles la bibliografía más acorde a sus características individuales.
- Se obtuvo una aplicación desarrollada a partir del uso de tecnologías y herramientas de software libre, contribuyendo así a la independencia tecnológica.
- 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, se recomienda:

- 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. VALENCIA-MOLINA, Tatiana, SERNA-COLLAZOS, Andrea, OCHOA-ANGRINO, Solanlly, CAICEDO-TAMAYO, Adriana María, MONTES-GONZÁLEZ, Jairo Andrés and CHÁVEZ-VESCANCE, José David. Competencias y estándares TIC desde la dimensión pedagógica: una perspectiva desde los niveles de apropiación de las TIC en la práctica educativa docente. *MINISTERIO DE EDUCACION* [online]. 2016. [Accessed 15 November 2018]. Available from: <http://repositorio.minedu.gob.pe/handle/123456789/4757>
2. Importancia de las TIC en el proceso de enseñanza aprendizaje. [online]. 15 November 2018. [Accessed 15 November 2018]. Available from: <http://www.educando.edu.do/articulos/docente/importancia-de-las-tic-en-el-proceso-de-enseanza-aprendizaje/>
3. PONTES-PEDRAJAS, Alfonso. Aplicaciones de las Tecnologías de la Información y de la Comunicación en la educación científica. Primera parte: funciones y recursos. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias* [online]. 2005. Vol. 2, no. 1. [Accessed 15 November 2018]. Available from: <http://www.redalyc.org/resumen.oa?id=92020102>
4. EL USO DE LAS PLATAFORMAS DE ENSEÑANZA VIRTUAL PARA IMPARTIR ASIGNATURAS JURÍDICAS. [online]. 16 November 2018. [Accessed 16 November 2018]. Available from: <http://www.eumed.net/rev/rejie/01/jmadg.htm>
5. Sistema Tutores Inteligentes. *EcuRed* [online]. 16 November 2018. [Accessed 16 November 2018]. Available from: https://www.ecured.cu/Sistema_Tutores_Inteligentes
6. FERNANDO J. LAGE, Zulma Cataldi. *Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia* [online]. 2010. Available from: liema@fi.uba.ar, flage@fi.uba.ar
7. LUIS ERNESTO MANCHÓN LAUZARDO, Lourdes Alicia Risco Ferrer. *Sistema tutor inteligente para la asignatura Reconocimiento de patrones*. Universidad de las Ciencias Informáticas, 2017.
8. Desarrollo y Evaluación de un Sistema Tutorial Inteligente para el Apoyo de la Enseñanza de la Lectura Inicial | Strasser | Psykhe. [online]. 1997. Available from: <http://www.psykhe.cl/index.php/psykhe/article/view/102>
9. DISEÑO DE HEDEA: UNA HERRAMIENTA PARA LA CONSTRUCCION DE SISTEMAS TUTORES INTELIGENTES. [online]. 16 November 2018. [Accessed 16 November 2018]. Available from:

http://scholar.googleusercontent.com/scholar?q=cache:kpgHGWEX6lsJ:scholar.google.com/+DISE%C3%91O+DE+HEDEA:+UNA+HERRAMIENTA+PARA+LA++CONSTRUCCION+DE+SISTEMAS+TUTORES++INTELIGENTES+++&hl=es&as_sdt=0,5

10. OLGA LIDIA PÉREZ GONZÁLEZ, Laura Casas Fuentes, YAILÉ CABALLERO MOTA, Lis andra Docampo, SABEL YORDI GONZÁLE, Lenniet Coello and ANGELA MARTÍN. *Sistema Inteligente para el Álgebra Lineal* [online]. Available from: <http://funes.uniandes.edu.co/4458/1/CasasSistemaALME2013.pdf>
11. ESTRADA, Barrón, LUCÍA, María, ZATARAIN CABADA, Ramón and HERNÁNDEZ PÉREZ, Yasmín. Tutor Inteligente con reconocimiento y manejo de emociones para Matemáticas. *Revista electrónica de investigación educativa*. 2014. Vol. 16, no. 3, p. 88–102.
12. Verónica Amela Tarongí. *Sistema Tutor Inteligente Adaptativo para Laboratorios Virtuales y Remotos*. Trabajo de Investigación. Universidad Politécnica de Valencia, 2010.
13. YARISLEYDI DE AVILA FERNÁNDEZ, Luisa Idorka Morales Rodríguez. *Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos*. Universidad de las Ciencias Informáticas, 2012.
14. JOSÉ ENRIQUE PACHECO PIÑEYRO, Lidiana Cruz Pazos. *Sistema de Enseñanza-Aprendizaje Inteligente de apoyo al curso introductorio de Matemática*. Universidad de las Ciencias Informáticas, 2010.
15. SÁNCHEZ, Natalia Martínez, LORENZO, María Matilde García and VALDIVIA, Zoila Zenaida García. Modelo para diseñar sistemas de enseñanza-aprendizaje inteligentes utilizando el razonamiento basado en casos. *Avances en Sistemas e Informática*. 1 September 2009. Vol. 6, no. 3, p. 67–78.
16. Dr. Daniel Gálvez Lio. *Sistemas Basados en el Conocimiento*. Departamento de Ciencia de la Computación en la Facultad de Matemática, Física y Computación de la Universidad Central “Martha Abreu” de Las Villas, [no date].
17. José Manuel Gutiérrez. *Sistemas Expertos Basados en Reglas* [online]. Dpto. de Matemática Aplicada. Universidad de Cantabria, [no date]. Available from: <http://personales.unican.es/gutierjm/cursos/expertos/Reglas.pdf>
18. Ing. José Joskowicz. *Reglas y Prácticas en eXtreme Programming* [online]. Available from: <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>

19. El Lenguaje de Modelado Unificado (UML). [online]. 3 December 2018. [Accessed 3 December 2018]. Available from: <https://www.docirs.com/uml.htm>
20. Visual Paradigm - UML, Agile, PMBOK, TOGAF, BPMN and More! [online]. 3 December 2018. [Accessed 3 December 2018]. Available from: <https://www.visual-paradigm.com/features/>
21. *PHP y MySQL* [online]. 2018. [Accessed 6 December 2018]. Available from: https://books.google.com/books/about/PHP_y_MySQL.html?hl=es&id=zMK3GOMOpQ4C
22. phpMyAdmin. [online]. [Accessed 19 March 2019]. Available from: <https://www.phpmyadmin.net/>
23. Adriel Hernández Lugo. *Base de datos para el Sistema de Gestión de Investigaciones*. La Habana, Cuba : Universidad de las Ciencias Informáticas, 2012.
24. MONTOYA, Carlos Eduardo Gómez, URIBE, Christian Andrés Candela and RODRÍGUEZ, Luis Eduardo Sepúlveda. Seguridad en la configuración del servidor web Apache. *INGE CUC*. 31 December 2013. Vol. 9, no. 2, p. 31–38.
25. Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto. [online]. [Accessed 16 May 2019]. Available from: https://netbeans.org/index_es.html
26. ZULMA, Cataldi and FERNANDO J., Lage. Herramienta para determinación de estilos de aprendizaje de los estudiantes y enfoques de la enseñanza. . 2008.
27. Katerine Villamizar Suaza. *Definición de equivalencias entre historias de usuario y especificaciones en UN-LENCEP para el desarrollo ágil de software*. Magister. Medellín, Colombia : Universidad Nacional de Colombia, 2013.
28. Daimara Mustelier Sanchidrian. *Método para determinar la complejidad de los requisitos funcionales de software* [online]. Máster. Universidad de las Ciencias Informáticas, 2014. Available from: https://repositorio_institucional.uci.cu/jspui/bitstream/ident/8552/1/Daimara%20Mustelier%20Sanchidrian-TM.pdf
29. Desarrollo de software. Tarjetas CRC. *Jummp* [online]. 10 January 2012. [Accessed 12 March 2019]. Available from: <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>

30. Eugenia Bahit. El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC. . 2011. P. 66.
31. Lic . Ariel Trellini. Principios de Diseño GRASP. Diseñando objetos con responsabilidad. [online]. Departamento de Ciencias e Ingeniería de la Computación. Universidad Nacional del Sur. 2015. Available from:
<http://cs.uns.edu.ar/~atrellini/ayds/downloads/Clases/18.%20AyDS%20-%20Principios%20GRASP.pdf>
32. Lic. Elisa Arizaca Ramirez. *Diagrama de despliegue* [online]. Available from:
<http://virtual.usalesiana.edu.bo/web/practica/archiv/ddespli2.doc>.
33. BEDREGAL, Velarde and RAÚL, Héctor. *Modelo para la estimación del esfuerzo de desarrollo en tareas de ingeniería de proyectos de software empleando aprendizaje automático* [online]. Universidad de Granada, 2017. [Accessed 17 March 2019]. ISBN 978-84-9163-136-1. Available from:
<http://digibug.ugr.es/handle/10481/45264>
34. Julián Andrés Mera-Paz. Análisis del proceso de pruebas de calidad de software. 2016. Vol. 12, p. 163–176. 20
35. Usando Jmeter para pruebas de rendimiento. [online]. [Accessed 26 May 2019]. Available from:
<http://sedici.unlp.edu.ar/handle/10915/21907>
36. F JAVIER, Diaz, CLAUDIA, Tzancoff, ANAHÍ, Rodríguez and VALERIA, Soria. Usando Jmeter para pruebas de rendimiento. [online]. 2008. [Accessed 26 May 2019]. Available from:
<http://sedici.unlp.edu.ar/handle/10915/21907>