



Universidad de las Ciencias Informáticas
Facultad 2

Título:

“Herramienta informática para el diagnóstico inicial en el Sistema
Tutorial Inteligente de Reconocimiento de Patrones.”

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

Autores:

Beatriz Lugo Robaina

Liomar Rodríguez Guerra

Tutores:

Msc. Yunia Reyes González

Ing. Maidelis Milanés Luque

“La Habana, junio del 2017”



*"Yo rechazo la mentira porque sé que la ignorancia
ha sido la gran aliada de la opresión a lo largo de la
historia."*

Fidel Castro

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

Para que así conste firmamos la presente a los ___ del mes de junio de 2017.

Beatriz Lugo Robaina

Liomar Rodríguez Guerra

Autor

Autor

Ing.Maidelis Milanés Luque

Mcs. Yunia Reyes Gonzáles

Tutor

Tutor

AGRADECIMIENTOS

*Agradezco especialmente a mis padres por su apoyo incondicional,
por no dudar ni un segundo de que este día llegaría a pesar de mis
constantes abatimientos.*

*A mi hermana Dayana, a Yasmany, a mi familia por su confianza
incondicional.*

*A Yulio por apoyarme desde el momento que me conoció, por
brindarme su amor incondicional y especialmente por darme fuerzas
para continuar.*

A mis suegros por acogerme como una hija más.

*A mi compañero de tesis por aceptar todos los regaños y las
sugerencias, por su comprensión y disposición para lograr las cosas.*

Muchas Gracias, Beatriz

AGRADECIMIENTOS

A mi mamá, a mi padrastro les agradezco grandemente por tanto sacrificio para lograr apoyarme en todo momento y saber comprender cada decisión que tomaba.

A mis amigos Niurka, Juan Pablo, Yulio, Duniel, Sady, Camilo, Yadian, Yasiel por darme su mano cuando más la necesitaba, a la hora que fuera en el momento que fuera.

A mis tutoras las cuales ya a final de curso soñaban hasta con nosotros, pero les agradezco cada minuto que pasaron apoyándonos.

También a todas mis amistades que de una forma u otras nos conocimos y están presentes aquí, son tantas que sería imposible mencionarlas a todas y no quiero dejar de mencionar a ninguna.

A todas esas personas que hicieron hasta lo imposible para que no me graduara, también tiene un espacio en mis agradecimientos, pues gracias a ellas me hicieron más fuerte.

A mi compañera de tesis le agradezco su paciencia, comprensión y amistad.

Muchas Gracias, Liomar

AGRADECIMIENTOS COMPARTIDOS

A nuestras tutoras Maidelis y Yunia por su atención, su preocupación, dedicación y pensamiento positivo.

A los compañeros del aula y los que una vez fueron, por compartir nuestro paso por la universidad.

A todos los profesores que han contribuido durante estos cinco años en nuestra formación como ingenieros.

A la revolución por brindarnos la posibilidad de convertirnos en ingenieros.

A todos que de una forma u otra han favorecido a la realización de este sueño.

DEDICATORIAS

*A mis padres por el amor, la confianza y el apoyo que siempre me
han brindado desde nací.*

BEATRIZ

*A mi mamá y a mi padrastro por nunca rendirse a pesar de las
malas noticias.*

LIOMAR

INDICE

INTRODUCCIÓN	1
1.1 Introducción	5
1.2 Modelado del estudiante en los Sistemas Tutoriales Inteligentes	5
1.3 Diagnóstico inicial en los Sistemas Tutoriales Inteligentes	6
1.4 Análisis de herramientas similares	7
1.5 Estilos de aprendizajes y tipos de inteligencias	12
1.6 Técnicas utilizadas	15
1.6.1 Reglas de producción	15
1.6.2 Agrupamiento Conceptual	16
1.7 Metodología de desarrollo de software	17
1.8 Tecnologías y herramientas seleccionadas	18
1.8.1 Tecnologías del lado del cliente	19
1.8.2 Tecnologías del lado del servidor	20
1.8.3 Entorno de Desarrollo Integrado	21
1.9 Sistema de Gestión de Base de Datos	21
1.10 Servidor web	22
1.11 Herramienta CASE	23
1.12 Lenguaje de modelado	23
1.13 Marco de trabajo	24
1.14 Conclusiones parciales	25
Capítulo 2: Diseño de la propuesta de solución	26
2.1 Introducción	26
2.2 Modelo de dominio	26
2.3 Propuesta de solución	27
2.4 Fase de exploración	30
2.4.1 Características funcionales del sistema	30
2.4.2 Especificación de las Historia de Usuarios	30

2.4.3 Características no funcionales del sistema	31
2.5 Fase de planificación	33
2.5.1 Plan de iteraciones	33
2.5.2 Plan de entrega	33
2.6 Fase de diseño del sistema	34
2.6.1 Estilo arquitectónico.....	34
2.6.2 Patrones de diseño	36
2.6.3 Diseño de la Base de Datos	37
2.6.4 Tarjetas CRC	38
2.7 Conclusiones parciales	39
3.1 Introducción	41
3.2 Implementación	41
3.2.1 Diagrama de despliegue.....	42
3.2.2 Estándares de codificación.....	43
3.3 Pruebas	43
3.3.1 Pruebas unitarias.....	44
3.3.2 Pruebas de aceptación.....	46
1.3.3 Pruebas de usabilidad.....	47
1.3.4 Pruebas de seguridad	49
3.4 Conclusiones parciales.....	50
CONCLUSIONES GENERALES	51
RECOMENDACIONES.....	52
REFERENCIAS BIBLIOGRÁFICAS.....	53
BIBLIOGRAFÍA.....	55
ANEXO 1.....	57
ANEXO 2.....	60
ANEXO 3.....	64
ANEXO 4.....	67

ÍNDICE DE TABLAS

Tabla 1 HU-5 Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia.	31
Tabla 2 HU-9 Agrupar estudiantes.	31
Tabla 3 Plan de iteraciones	33
Tabla 4 Plan de entrega	34
Tabla 5 Tarjeta CRC DataProcessed	38
Tabla 6 Tarjeta CRC InitSurvySTI_LC_Conceptual	38
Tabla 7 Tarjeta CRC LCConceptual_alg.....	38
Tabla 8 Tarjeta CRC MatrizBinaria	39
Tabla 9 Tarjeta CRC Testores_alg	39
Tabla 10 Tarjeta CRC FastBR.....	39
Tabla 11 Tarjeta CRC ConnectionDB.....	39
Tabla 12 Tareas de ingeniería.....	41
Tabla 13 Tarea de ingeniería: Analizar cuestionario.....	42
Tabla 14 Tarea de ingeniería: Clasificar estudiante	42
Tabla 15 Caso de Prueba de Aceptación 9	46
Tabla 16 Cantidad de no conformidades por cada iteración de las pruebas de aceptación.	47
Tabla 17 Cumplimiento de la lista de chequeo de usabilidad.....	48
Tabla 18 Resultados de la aplicación de la lista de chequeo para pruebas de seguridad	49
Tabla 19 Tarea de ingeniería: Cálculo de la Matriz de Diferencia	64
Tabla 20 Tarea de ingeniería: Cálculo de la Matriz Básica	64
Tabla 21 Tarea de ingeniería: Selección de los testores.	64

Tabla 22 Tarea de ingeniería: Calcular relevancia de los rasgos.....	65
Tabla 23 Tarea de ingeniería: Selección de los testores más relevantes.....	65
Tabla 24 Tarea de ingeniería: Generación de los I-complejos de cada grupo.....	65
Tabla 25 Caso de Prueba de Aceptación 13.	67

ÍNDICE DE FIGURAS

Figura 1 Componentes básicos del módulo del estudiante (Cataldi, 2012).	6
Figura 2 Modelo del dominio	27
Figura 3 Propuesta de solución (Fase II tomada de Tesis de Maestría de Yunia Reyes)	27
Figura 4 Clases del modelo.....	35
Figura 5 Clases de la vista	35
Figura 6 Clases del controlador	35
Figura 7 Modelo de la Base de Datos	37
Figura 8 Diagrama de despliegue	42
Figura 9 Comportamiento de las no conformidades por cada iteración de las pruebas de aceptación.	47
Figura 10 Cumplimiento de los indicadores de la lista de chequeo	48
Figura 11 Resultado final de la prueba de usabilidad	49
Figura 12 Datos de los expertos entrevistados.....	59

RESUMEN

El desarrollo de las Tecnologías de las Informáticas y las Comunicaciones (TIC), ha tenido gran influencia en las diversas esferas sociales, destacándose su uso en la educación, incorporando elementos novedosos que han revolucionado los métodos de enseñanza-aprendizaje. En la Universidad de las Ciencias Informáticas (UCI), se desarrolló un Sistema Tutorial Inteligente para la asignatura optativa Reconocimiento de Patrones, con el objetivo de apoyar el proceso de formación de los estudiantes en esa materia. El estado de conocimiento del estudiante se realiza a través de la asignación y evaluación de protocolos pedagógicos. Estos protocolos deben ser diseñados a partir de un diagnóstico inicial, previamente realizado al alumno.

El objetivo del presente trabajo de diploma, lo comprende el desarrollo de una herramienta informática para el diagnóstico inicial de los estudiantes en el Sistema Tutorial Inteligente de Reconocimiento de Patrones. Entiéndase por diagnóstico inicial la clasificación de los estudiantes según los estilos de aprendizaje y tipo de inteligencia a través de reglas definidas por expertos, y posteriormente el agrupamiento de ellos según el nivel de semejanza, utilizándose el algoritmo LC Conceptual.

Palabras claves: algoritmo LC Conceptual, estilos de aprendizaje, proceso de enseñanza aprendizaje, tipo de inteligencia, reconocimiento de patrones.

INTRODUCCIÓN

Las nuevas tecnologías han incrementado de forma gradual la utilización de los medios tecnológicos en diversas esferas de la sociedad, incluyendo la educación. Estas juegan un papel fundamental y una gran oportunidad para el perfeccionamiento de los profesores; ya que ofrecen facilidades al estudiante de acceder a la información, así como establecer el diálogo que le permite transformar la información en conocimiento (Martínez, 2014). La inserción de las Tecnologías de la Información y las Comunicaciones (TIC) en el ámbito educativo, plantea nuevos escenarios que requieren una revisión y actualización constante de las prácticas pedagógicas, especialmente en la educación universitaria.

Con relación al uso de las tecnologías en la educación, la Organización de las Naciones Unidas para la Educación y la Cultura (UNESCO, 2014), señala que es necesaria la aplicación efectiva de las herramientas que nos ofrece Internet para acceder a la información de manera organizada y con intencionalidad. También existen otras herramientas como son los recursos multimedia, los videos tutoriales, los Entornos Virtuales de Enseñanza Aprendizaje (EVEA), los softwares educativos, simuladores virtuales y los Sistemas Tutoriales Inteligentes (STI) (Argueta, 2012).

Los Sistemas Tutoriales Inteligentes (STI) son herramientas educativas que portan conocimientos sobre cierta materia. El propósito es transmitir este conocimiento a los estudiantes mediante un proceso interactivo individualizado, intentando simular la forma en que un profesor guiará al alumno en el proceso de enseñanza-aprendizaje. Generalmente están estructurados en tres componentes principales: módulo del tutor, módulo de dominio y módulo o modelado del estudiante. La principal característica que hace al sistema tutor inteligente es su capacidad para adaptarse a los estudiantes, por este motivo, el problema del modelado del alumno es un tema crítico, ya que es el núcleo del sistema (Cataldi, 2012).

El modelado del estudiante es el proceso encargado de representar al alumno en el sistema. Está estructurado por dos componentes fundamentales: estilos del estudiante y el estado de conocimiento, siendo el diagnóstico¹ el mecanismo que manipula esta estructura. Se considera que una de las cuestiones más importantes dentro del modelado del alumno es la selección adecuada de las características que han de

¹ Se refiere a la evaluación que realiza el estudiante para identificar tanto las características iniciales, como para comprobar los conocimientos adquiridos.

quedar reflejadas en el primer componente, las que deben ser identificadas antes de empezar el proceso de enseñanza-aprendizaje. Un diseño apropiado de este componente permitirá una mejor efectividad de la interacción del estudiante con el sistema, siendo este el diagnóstico inicial.

En la Universidad de las Ciencias Informáticas se imparte la asignatura optativa de “Reconocimiento de Patrones”, para la cual se desarrolló un Sistema Tutorial Inteligente. La herramienta se centra en la creación, asignación y evaluación de protocolos pedagógicos², para manejar el estado de conocimientos de los estudiantes. Considerando como entrada un estudiante con una clase previamente definida (clase 1, ..., clase N), sin embargo, no cuenta con un mecanismo que permita realizar esta clasificación inicial.

Se debe destacar que el sistema hoy asigna los protocolos pedagógicos a los estudiantes, independientemente si estos presentan un alto grado de semejanza entre ellos. Lo que puede resultar ineficiente cuando se esté en presencia de un número elevado de usuarios interactuando con el mismo.

Después de lo antes expuesto se evidencia que se requiere de un componente para diagnóstico inicial, que contemple la identificación en los estudiantes de los rasgos cognitivos previamente seleccionados. Así como la necesidad de evitar la duplicidad de información de los protocolos, a través de un agrupamiento conceptual para conformar los grupos y su correspondiente descripción.

Teniendo en cuenta la situación problemática anteriormente descrita, se plantea como **problema de investigación**: El diagnóstico inicial como parte del modelado del estudiante en el STI para la asignatura Reconocimiento de Patrones, en la Universidad de las Ciencias Informáticas.

Teniendo en cuenta el problema planteado se define como **objeto de estudio**: El proceso del modelado del estudiante en los Sistemas Tutoriales Inteligentes.

Se propone como **objetivo general**: Desarrollar una herramienta informática para realizar el diagnóstico inicial del estudiante en el Sistema Tutorial Inteligente para Reconocimiento de Patrones.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

² Protocolo pedagógico: método de enseñanza más acorde a las características del estudiante.

- Fundamentar los elementos asociados al modelado del estudiante en los Sistemas Tutoriales Inteligentes y el diagnóstico inicial como componente fundamental dentro de este.
- Definir las tecnologías, metodología y herramientas a utilizar para el desarrollo de la herramienta.
- Diseñar una herramienta informática para la realización del diagnóstico inicial a los estudiantes que inician el curso Reconocimiento de Patrones.
- Implementar la herramienta informática diseñada, haciendo un uso correcto del lenguaje y las técnicas a utilizar.
- Realizar pruebas para validar el correcto funcionamiento de la herramienta implementada.

El **campo de acción** se enmarca en: El proceso de diagnóstico inicial del estudiante en el Sistema Tutorial Inteligente para el Reconocimiento de Patrones.

Para dar cumplimiento a los objetivos planteados, se proponen realizar las siguientes **tareas de investigación**:

- Interpretación de los elementos asociados al modelado del estudiante y el diagnóstico inicial en los Sistemas Tutorial Inteligente.
- Análisis del componente de diagnóstico inicial en el modelado del alumno de diferentes Sistemas Tutorial Inteligente con el propósito de definir los elementos a tener en cuenta para el desarrollo de la herramienta.
- Definición de los elementos a tener en cuenta en el diagnóstico inicial del STI.
- Análisis de las técnicas de Inteligencia Artificial más utilizadas para la clasificación de los estudiantes.
- Análisis de la metodología de desarrollo de software que le sirva de guía para el proceso de elaboración de la herramienta.
- Análisis de las tecnologías, herramientas y lenguajes que le permitan el desarrollo de la propuesta de solución.
- Diseño del sistema a desarrollar, generando los artefactos de Ingeniería de Software correspondientes de acuerdo a la metodología de desarrollo, patrones y arquitectura a utilizar.
- Obtención del código fuente de la herramienta informática para el diagnóstico inicial.
- Ejecución de las pruebas de software para verificar el correcto funcionamiento de la propuesta de solución.

Para apoyar el desarrollo de la investigación se emplean los siguientes métodos científicos:

Métodos teóricos:

Análisis – síntesis: Empleado para el análisis de los elementos esenciales referentes a las teorías, documentos y literatura en general relacionada con el modelado del estudiante en los Sistemas Tutoriales Inteligentes, específicamente en el diagnóstico inicial.

Modelación: Utilizado en el diseño de la propuesta de solución y los artefactos generados por la metodología seleccionada.

Métodos empíricos:

Entrevista: Fue realizada a profesores con experiencia científica y académica que ostentan categoría científica y grado académico, para seleccionar los rasgos cognitivos³ a tener en cuenta en el diagnóstico inicial. También es utilizada para definir las características funcionales del sistema con el cliente.

Estructura metodológica del documento

Capítulo 1. Fundamentación teórica: Se fundamenta el modelado del estudiante en los STI, haciendo énfasis en el componente diagnóstico inicial. Se aborda la entrevista realizada en la definición de los rasgos cognitivos que se deben tener en cuenta para la clasificación del estudiante. Además, se analizan la metodología, las tecnologías, herramientas y lenguajes necesarios para la propuesta de solución.

Capítulo 2. Diseño de la propuesta de solución: Se explica la propuesta de solución, las características funcionales y no funcionales del sistema. Se especifican los artefactos generados según la metodología utilizada en su fase de exploración y planificación. Además, se detallan los patrones de diseño, arquitectura y diseño de la base de datos que sustentan la propuesta de solución.

Capítulo 3. Implementación y Prueba de la herramienta informática: Se especifican los artefactos generados según la metodología utilizada en su fase de implementación. Se evidencian los resultados de las pruebas unitarias para las funcionalidades, las pruebas de aceptación para las historias de usuario, así como las pruebas de usabilidad y seguridad realizadas a la herramienta.

³ Características que están relacionadas al conocimiento. Éste, a su vez, es el cúmulo de información que se dispone gracias a un proceso de aprendizaje o a la experiencia.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se analiza el modelado del estudiante en los Sistemas Tutoriales Inteligentes, haciendo énfasis en el componente diagnóstico inicial. Se exponen los resultados del análisis de diversos STI. Se definen los estilos de aprendizaje y tipos de inteligencias a evaluar en el diagnóstico, y las técnicas a utilizar para dar solución al problema planteado. Por último, se analiza la metodología, lenguaje y herramientas seleccionadas para el desarrollo del sistema.

1.2 Modelado del estudiante en los Sistemas Tutoriales Inteligentes

Zulma Cataldi plantea que el módulo del estudiante tiene como objetivo realizar el diagnóstico cognitivo del alumno, y el modelado del mismo para una adecuada retroalimentación del sistema. El módulo estudiante está estructurado por los siguientes submódulos (Cataldi, 2012).

- a) **Estilos de aprendizaje:** Está 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.
- b) **Estado de conocimientos:** Contiene el mapa de conocimientos obtenido inicialmente a partir del módulo del dominio y que el actualizador de conocimientos irá modificando progresivamente, a través de los resultados obtenidos en las evaluaciones efectuadas por el módulo del tutor, quién le enviará dichos resultados procesados.
- c) **Perfil psico-sociológico del estudiante:** Para determinar el perfil psico-sociológico se usa la Teoría de las Inteligencias Múltiples de Gardner.

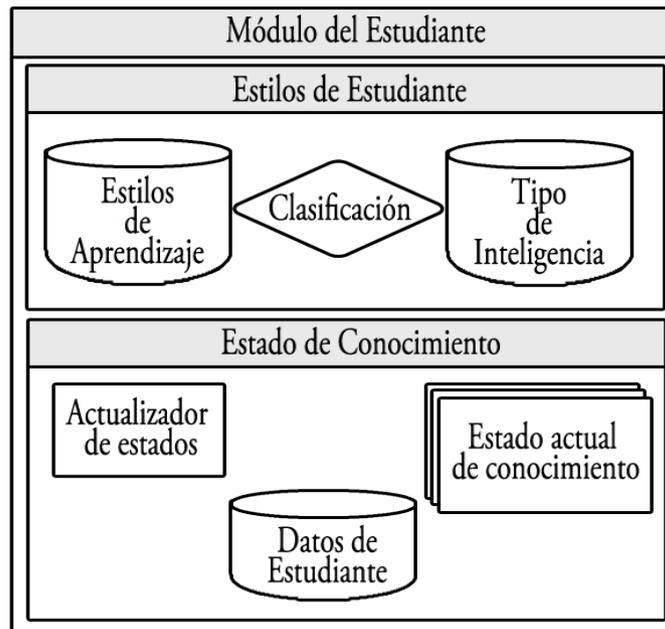


Figura 1 Componentes básicos del módulo del estudiante (Cataldi, 2012).

Como se muestra en la figura 1, el módulo del estudiante está conformado por dos submódulos: estilos de estudiante y estado de conocimiento, dentro de estilos de estudiante se clasifica los estilos de aprendizaje y tipo de inteligencia, considerándose este submódulo como diagnóstico inicial.

1.3 Diagnóstico inicial en los Sistemas Tutoriales Inteligentes

El diagnóstico no es más que el análisis de las características de un objeto o sistema a partir de las cuales se procede a realizar un tratamiento sobre determinado ámbito. El objetivo principal del diagnóstico inicial es identificar de cada estudiante los estilos de aprendizaje y su perfil psico-sociológico, también se puede considerar parte de este, el estado de conocimiento inicial del estudiante al interactuar con el STI por primera vez.

Un estilo de aprendizaje es la forma de clasificar el comportamiento de un estudiante de acuerdo a la manera en que toma la información, forma las estrategias para aprender, cómo entiende y cómo le gusta analizar la información que está utilizando para llegar a un conocimiento determinado. En otras palabras, es una forma de clasificar un estudiante de acuerdo a un perfil en relación con la información, ya que este estilo evoluciona y cambia de acuerdo a las variables de entorno y ambientales que lo afectan (Cataldi, 2012).

Para determinar el perfil socio - psicológico de cada estudiante se categoriza el tipo de inteligencia que posee. A través de los tipos de inteligencias se analizan las

capacidades cognitivas humanas, estas funcionan juntas para resolver problemas cotidianos, crear productos o para ofrecer servicios dentro del mismo ámbito cultural. Para determinar ambas características, los alumnos deben someterse a una serie de situaciones que al responder los podrán clasificar según el estilo de aprendizaje y el o los tipos de inteligencias, para posteriormente trazar una estrategia educativa que sea capaz de apoyarlos en su desempeño en el curso y desarrollar una atención diferenciada atendiendo a sus debilidades y fortalezas.

1.4 Análisis de herramientas similares

En una investigación el análisis de herramientas similares, permite la identificación de elementos que puedan ser de utilidad para el desarrollo adecuado de la misma.

Para la selección de las herramientas que se debían analizar se siguieron los siguientes criterios:

1. Que fueran considerados sistemas tutores inteligentes.
2. Que dentro del modelado del estudiante realizaran diagnóstico inicial, para analizar los rasgos cognitivos reflejados y las técnicas utilizada para identificarlos.

ELE-Tutora: Un Sistema Tutorial Inteligente para la focalización de los errores gramaticales en la enseñanza del español como Lengua Extranjera, 2015.

ELE-Tutora fue desarrollado por Anita Ferreira de la Universidad Concepción de Chile en 2015. Tiene por objetivo apoyar al estudiante en el nivel de proficiencia B2 (acorde con el Marco Común Europeo de Referencia para las lenguas o MCER (Consejo de Europa, 2002)) que se entiende como un nivel intermedio-avanzado, pues considera entre sus competencias comunicativas la comprensión y producción de textos complejos, tanto orales como escritos sin mayor esfuerzo.

Este es un sistema tutorial inteligente en lo que compete a la interacción con el estudiante, el tratamiento de los errores de lengua, la retroalimentación y adaptación a los diferentes niveles de proficiencia. Está constituido por los tres módulos básicos de un STI, el módulo del estudiante contiene información acerca del estudiante (su conocimiento del dominio, su nivel de proficiencia en el español como lengua extranjera, sus errores gramaticales más frecuentes, su estilo de aprendizaje, etc) (Ferreira, 2015).

ELE-Tutor Sistema Tutorial para el español como Lengua Extranjera. (Universidad de Concepción, Chile), 2012.

ELE Tutor es un STI para la enseñanza del español como lengua extranjera con el objetivo de lograr adaptar el sistema a la necesidad de cada estudiante, fue desarrollado por Anita Ferreira y Pedro Salcedo en la Universidad de Concepción de Chile en el 2012.

El modelo del estudiante recoge una serie de información con respecto a cada estudiante que interactúa con él. En el modelo planteado, el sistema recibe información con respecto al estilo de aprendizaje, el nivel de proficiencia y el tipo de error que comete el estudiante, lo cual permite que el módulo tutor genere una retroalimentación específica y adaptada a las características de cada estudiante.

La información referente al estilo de aprendizaje de los estudiantes se obtiene mediante un test de entrada, el *Embedded Figures Test* (EFT), diseñado específicamente para determinar la tendencia del estudiante en cuanto a la polaridad dependencia/independencia de campo (Witkin, 1962) y el cual es aplicado al estudiante por el mismo STI antes de que el estudiante comience con los ejercicios propuestos (Ferreiro, 2012).

En la investigación no queda reflejado si usan alguna técnica de Inteligencia Artificial para analizar los resultados del test .

TuinLEC, tutor inteligente para mejorar la competencia lectora, 2014.

TuinLEC es un STI desarrollado por el grupo Psicotext de la Universidad de Valencia, España, en 2014 se crea con la finalidad de enseñar y mejorar estrategias de competencia lectora en alumnos de 6º de Educación Primaria. El tutor permite controlar y registrar con exactitud la secuencia de lectura y los tiempos de la misma, a la vez que se simula la lectura natural en pantalla de ordenador. Este sistema permite ir más allá de los datos clásicos de producto (puntuación final de comprensión) y avanzar en el conocimiento de estrategias de comprensión (cantidad de texto leído inicialmente, decisiones de relectura, búsqueda de información relevante para responder, etc.).

Antes de interactuar con TuinLEC a los estudiantes se les aplica el Test de Estrategias de Comprensión, en su versión electrónica para medir la capacidad general de comprensión de cada estudiante como pre-test, siendo esto su forma de realizar el

diagnóstico inicial. Registra los datos on-line del proceso de lectura y de contestación a preguntas mediante la técnica de ventana móvil, a través de la cual los estudiantes leen los textos de forma secuencial mientras son presentados en una ventana del ordenador (Psicotext, 2014).

En la investigación no se tiene constancia si identifican de los elementos que define Zulma Cataldi en el modelado del estudiante, no queda reflejado si usan alguna técnica de Inteligencia Artificial para clasificar estudiantes a través del Test de Estrategias de Comprensión.

Tutor Inteligente con reconocimiento y manejo de emociones para Matemáticas, 2014.

El tutor fue desarrollado por María Lucía Barrón, Ramón Zatarain y Yasmin Hernández, es una colaboración entre el Instituto Tecnológico de Culiacán Departamento de Ciencias de la Computación y el Instituto de Investigaciones Eléctricas Gerencia de Tecnologías de la Información en México en el 2014. Es un sistema 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.

Los usuarios de este tutor tienen asociados información personal, académica y afectiva en un perfil, el cual se obtiene y almacena en forma estática y dinámica. Utilizan como diagnóstico inicial la recogida de información para el perfil estático que contiene datos del usuario (por ejemplo, información personal y académica), mientras que el perfil dinámico se actualizará de acuerdo a la interacción del usuario dentro de la red y del STI, tomando en cuenta los diferentes aspectos sociales, cognitivos y emocionales.

El módulo del estudiante es responsable de evaluar el desempeño para determinar habilidades cognitivas y de razonamiento. Provee la información sobre las competencias y capacidades de aprendizaje del estudiante. Es identificado el conocimiento del estudiante a través de un examen diagnóstico inicial, además de utilizar para el reconocimiento de emociones una red neuronal artificial (Zatarain, 2014). Pero se necesita la disponibilidad de dispositivos que permitan reconocimiento de rostro en tiempo real.

Sistema Tutorial Inteligente para el Apoyo de la Enseñanza de la Lectura Inicial, 2000.

El sistema fue desarrollado por Katherine Strasser y Ricardo Rosas de la Pontificia Universidad Católica de Chile, es un software educativo para apoyar el proceso de aprendizaje basado en los conocimientos actuales sobre la enseñanza de la lectoescritura y en el modelo de Sistemas Instruccionales Inteligentes.

En el módulo del estudiante se representan las diversas características de los alumnos en relación al dominio, como por ejemplo su nivel de conocimiento, sus errores más frecuentes, sus concepciones erróneas, su velocidad de aprendizaje, etc. El sistema contiene actividades instruccionales y evaluativas, evalúa al estudiante a través de juegos (Strasser, 2000).

En la investigación no se refleja de la utilización de un diagnóstico inicial para identificar los elementos que define Zulma Cataldi en el modelado del estudiante, no queda plasmado si usan alguna técnica de Inteligencia Artificial para clasificar estudiantes.

Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos, 2012.

El sistema fue desarrollado por Luisa Idorka Morales y Yarisleidy Ávila de la facultad 2 en la Universidad de las Ciencias Informáticas en el 2012. Tiene como objetivo apoyar el proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos (SO) para la Facultad 2 y ayudar a consolidar los contenidos del plan de estudio definido para la asignatura.

El STI posee un conjunto de cuestionarios con el objetivo de captar el estado cognitivo de los estudiantes y a partir de esto definir los materiales de estudio de cada uno. El módulo del estudiante está compuesto por los rasgos predictores formados por las preguntas (Morales, 2012).

Actualmente el STI no se encuentra en explotación, además de no reflejan la utilización de alguna técnica de Inteligencia Artificial para clasificar estudiantes.

Ambiente inteligente para el aprendizaje del lenguaje estructurado de consulta.

El STI fue desarrollado por Yaritza Gonzáles y Osiel Menéndez de la Facultad 4 de la Universidad de las Ciencias Informáticas. El sistema se desarrolló con el objetivo de

apoyar el aprendizaje del lenguaje estructurado de consulta en la asignatura Sistema de Bases de Datos, que permita realizar la corrección de los ejercicios realizados por los estudiantes y a la vez brindar información sobre los errores cometidos. Consta con una sesión destinada a los estudiantes que serán los encargados de dar respuesta a los problemas publicados y tendrán además la posibilidad de desarrollar otras acciones.

Actualmente el STI no se encuentra en explotación, no se evidencia la realización de un diagnóstico inicial para identificar los elementos que define Zulma Cataldi en el modelado del estudiante. A pesar de que utilizan como técnica de Inteligencia Artificial Sistemas Basados en Reglas, no lo hacen para la clasificación inicial.

Sistema Inteligente para el Álgebra Lineal, 2012.

El Sistema Inteligente para el Álgebra Lineal fue creado por Laura Casas Fuentes y Olga Lidia Pérez en la Facultad de Informática, Universidad de Camagüey con el objetivo de apoyar al estudiante durante su estudio independiente para brindarle la vía de solución de los problemas que debe resolver.

Los fundamentos teóricos de la investigación 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 además de utilizar el Razonamiento basado en Casos (Pérez y otros, 2012).

Actualmente el STI no se encuentra en explotación, además que en la investigación no se evidencia la realización de un diagnóstico inicial para identificar los elementos que define Zulma Cataldi en el modelado del estudiante. A pesar de que utilizan como técnica de Inteligencia Artificial Sistemas Basados en Reglas, no lo hacen para la clasificación inicial.

Tutor Virtual para el aprendizaje Autónomo de Idiomas, 2010.

El tutor se encarga de facilitar las bases necesarias para el aprendizaje autónomo de idiomas, fue desarrollado por Danirys Ortega y Reinaldo Rodríguez de la Universidad de las Ciencias Informáticas.

Lleva a cabo el procesamiento de la información a partir de un conjunto de encuestas iniciales respondidas por el usuario, además de determinar el estado actual del mismo, así como su estilo o estilos de aprendizaje a la vez que genere la ruta de

autoaprendizaje a seguir durante el entrenamiento mejorando la calidad en el proceso de formación académica en esta disciplina (Ortega, 2010).

Actualmente el STI no se encuentra en explotación, no se refleja la aplicación de alguna técnica de Inteligencia Artificial para clasificar inicialmente a los estudiantes.

Resultados del estudio

- Al analizar las herramientas se puede afirmar que tienen en cuenta los estilos de aprendizajes como característica a identificar en el diagnóstico inicial.
- El diagnóstico inicial es realizado a través de cuestionarios, pero se desconoce cómo son procesados los datos.
- Ninguna de las herramientas realiza agrupamiento conceptual.
- Las herramientas de la UCI no se encuentran en explotación.

1.5 Estilos de aprendizajes y tipos de inteligencias

En las fuentes bibliográficas consultadas se evidencia la diversidad de clasificaciones de estilos de aprendizaje según diferentes autores: Kolb, Silver y Felderman, Honey y Mumford, entre otros. Debido a esto fue necesario realizar una entrevista para definir cuales conformarían el diagnóstico inicial. En el caso de los tipos de inteligencia después de realizar una amplia investigación se constató que la teoría más referenciada es la propuesta por Gardner, por lo que la entrevista fue confeccionada sobre un único autor.

Teniendo en cuenta como premisa que la UCI solo tiene 15 años de fundada, donde no predominan las categorías docentes y científicas superiores. La selección de los expertos se encuentra sustentada según los siguientes criterios, [ver Anexo 1:](#)

1. Profesores que pertenecieran al CICE.
2. Profesores que tuvieran 10 o más años de experiencia docente en carreras de corte ingenieril.
3. Profesores que hayan impartido al menos por 3 cursos la asignatura Reconocimiento de Patrones.

Para diseñar la entrevista se manejaron las diferentes clasificaciones propuestas por autores antes mencionados.

Los Estilos de Aprendizaje según Kolb (Agudelo, 2010)

Kolb (1984) incluye el concepto de estilos de aprendizaje dentro de su modelo de aprendizaje por la experiencia y lo describe como "algunas capacidades de aprender, que se destacan por encima de otras, como resultado del aparato hereditario de las experiencias vitales propias, y de las exigencias del medio ambiente actual".

Kolb combinó las dos dimensiones del aprendizaje (dimensión abstracta-concreta y activa-reflexiva) y encontró que las personas se sitúan en cuatro estilos de aprendizaje: convergente, divergente, asimilador y acomodador.

- Estilo Convergente
- Estilo Divergente
- Estilo Asimilador
- Estilo Acomodador

Los Estilos de Aprendizaje según P. Honey y A. Mumford (Ospina, 2010)

Honey y Mumford (1986) han partido del análisis de la teoría de Kolb, para llegar a una aplicación de los estilos de aprendizaje. Les preocupa averiguar por qué en una situación, en la que dos personas comparten texto y contexto, una aprende y otra no.

Los estilos de aprendizaje, para Honey y Mumford son también cuatro, que a su vez son las cuatro fases de un proceso cíclico de aprendizaje.

- Activo
- Reflexivo
- Teórico
- Pragmático

Los Estilos de Aprendizaje según Felder y Silverman (Ospina, 2010)

El modelo de Felder y Silverman clasifica los estilos de aprendizaje a partir de cinco dimensiones:

- Sensitivo/Intuitivo
- Visual/ Verbal
- Secuencial/Global
- Activo/Reflexivo

Modelo de Bandler y Grinder (Krusche,1996)

Este modelo, también llamado visual-auditivo-kinestésico (VAK), toma en cuenta el criterio neurolingüístico, que considera que la vía de ingreso de la información (ojo, oído, sensación) resulta fundamental en las preferencias de quien aprende o enseña. Presenta tres sistemas para representar mentalmente la información: el visual, el auditivo y el kinestésico.

- Visual
- Auditivos
- Kinestésico

También existen otros tipos de estilos de aprendizajes tales como:

- Social
- Auditivo/Musical
- Individual

Teoría de las inteligencias múltiples de Gardner (Gardner, 1998):

- Inteligencia Lingüística
- Inteligencia Lógico-matemática
- Inteligencia Espacial
- Inteligencia Corporal- kinestésica
- Inteligencia Interpersonal
- Inteligencia Intrapersonal
- Inteligencia Naturalista

Resultados de la entrevista

Al procesar los resultados de la entrevista se concluyó:

- Más del 60% de los expertos consideran que de los estilos de aprendizaje se deben identificar: activo/ reflexivo, visual/verbal y teórico,
- El 100% de los expertos está de acuerdo que se debe trabajar con la inteligencia lógico matemática.

Estilo Activo/Reflexivo: Los estudiantes procesan la información mediante tareas activas a través de compromisos en actividades físicas, discusiones o a través de la reflexión. (Propuesto por Felder y Silverman)

Estilo Visual/ Verbal: Los estudiantes procesan la información en formatos visuales: cuadros, diagramas, gráficos, demostraciones, etc. o en formatos verbales: sonidos, expresión oral y escrita, fórmulas, símbolos, etc. (Propuesto por Felder y Silverman)

Estilo Teórico: Los estudiantes procesan la información cuando se les presenta como parte de un sistema, modelo, teoría o concepto. Enfocan los problemas de forma escalonada, por etapas lógicas. (Propuesto por P. Honey y A. Mumford)

Inteligencia lógica matemática: Es la capacidad para usar los números de manera efectiva y de razonar adecuadamente. Incluye la sensibilidad a los esquemas y relaciones lógicas, las afirmaciones y las proposiciones, las funciones y otras abstracciones relacionadas.(Gardner)

Una vez seleccionados los rasgos cognitivos a tener en cuenta, se decide utilizar cuestionarios propuestos por el Centro de Innovación y Calidad de la Educación (CICE) de la Universidad de las Ciencias Informáticas, para identificar el estilo de aprendizaje visual/verbal y la inteligencia lógico matemática. En el caso del resto de los estilos de aprendizaje se utiliza la propuesta implementada por los propios autores [ver Anexo 2.](#)

1.6 Técnicas utilizadas

De acuerdo a la problemática planteada se decidió utilizar reglas para clasificar a los estudiantes según los rasgos identificados, y el algoritmo LC Conceptual para conformar los grupos de estudiantes con características similares.

1.6.1 Reglas de producción

Una regla de producción consta de un par ordenado (A,B), representado en el cálculo proposicional como $A \Rightarrow B$, donde A representa el antecedente y B el consecuente de la regla.

Una regla de producción se interpreta de la siguiente manera: si se satisface el antecedente, entonces se cumple el consecuente. Esta manera de interpretar una

regla permite considerarla como una unidad relativamente independiente de conocimiento.

Las reglas de producción pueden adoptar varias formas:

- a) Si condición P entonces conclusión C.
- b) Si situación S entonces acción A.
- c) Si condición C1 entonces no condición C2.

1.6.2 Agrupamiento Conceptual

El agrupamiento conceptual tiene como objetivo estructurar grupos a partir de una colección de objetos, proveer el significado de esas agrupaciones y aportar información sobre el sentido que tiene que los objetos pertenezcan a un mismo grupo, brinda las características o conceptos para esas agrupaciones que se definen sobre la base de los rasgos en términos de los cuales se describen a esos objetos que conforman los grupos.

El algoritmo LC Conceptual es un algoritmo de agrupamiento que tiene la capacidad de operar tanto con atributos cualitativos como cuantitativos, además de admitir la ausencia de información y no requiere especificar el número de agrupamientos a priori. El agrupamiento se realiza a través de la semejanza entre objetos, y los conceptos construidos son propiedades lógicas apoyadas en los rasgos que describen a los objetos en análisis, estos conceptos son fáciles de interpretar por los usuarios y facilita comprender las características fundamentales.

Debido a sus características, este algoritmo permite resolver una gama amplia de problemas de estructuración conceptual que otros algoritmos no pueden resolver satisfactoriamente, aunque la etapa de caracterización que ejecuta para formar los conceptos es computacionalmente costosa, principalmente debido al cálculo de los testores, la determinación de la descripción extensional de los grupos es computacionalmente poco costosa.

Descripción del algoritmo

Sea una colección de objetos, descritos en términos de un conjunto de atributos que pueden ser cualitativos, numéricos o incluso puede desconocerse su valor. Sea además S una función tal que, establece el grado de semejanza que tiene un par de objetos $O_i, O_j \in O$. El algoritmo LC representa la colección de objetos como un grafo

en el cual los objetos de la colección son los vértices del grafo y existe una arista entre dos objetos O_i y O_j , si donde β es un número real en el intervalo $[0, 1]$ que determina el umbral mínimo de semejanza que deben tener dos objetos para ser considerados como semejantes.

A partir del grafo anterior, los grupos son aquellos subconjuntos de O que satisfacen un conjunto de propiedades relativas a las semejanzas entre los objetos que constituyen cada subconjunto; estas propiedades se conocen como criterio de agrupamiento. El algoritmo LC propone dos criterios de agrupamiento que forman agrupamientos disjuntos. Uno de estos criterios forma grupos que son componentes β -conexas y el otro, grupos que son conjuntos β -compactos.

A partir de los grupos formados se construyen las propiedades (conceptos) que caracterizan a cada grupo de objetos. Se forma una matriz de aprendizaje MA a partir de MI y de los K_1, \dots , luego se calcula el conjunto de los testores más relevantes de MA $\tau = \{ \tau_1, \dots, \tau_j \}$.

Para cada clase K_i , $i=1, \dots$, se calcula la estrella $G(K_i \setminus K_1, \dots, K_{i-1}, K_{i+1}, \dots, K_c)$ que estará formada por p l-complejos (tantos como testores relevantes). Para el cálculo de los $R \mid K_i$ en cada $X \in \tau_q$, $\tau_q \in \tau$ se emplea el operador de refusión condicionada RUC. Los conceptos que caracterizan a cada agrupamiento K_i serán los l-complejos obtenidos en el paso anterior. (Reyes González, 2014).

Para el desarrollo del sistema se definió: metodología de software, tecnologías del lado del cliente, tecnologías del lado del servidor, lenguaje de programación, sistema gestor de Base de Datos, servidor web y marco de trabajo.

1.7 Metodología de desarrollo de software

Para la selección de la metodología se utilizó la estrella de Boehm y Turner, la cual arrojó que en dependencia de las características que posee el proyecto lo más idóneo es utilizar un criterio ágil, por lo que se seleccionó la metodología XP.

Xtreme Programming (XP): La programación extrema es una metodología ligera, iterativa incremental, creada para desarrollar software en equipos pequeños y medianos que trabajan en proyectos con requerimientos difusos o cambiantes. Se realizan pruebas constantemente del sistema y se le ofrece al cliente versiones continuas del mismo, esto permite que se consigan productos usables con mayor rapidez que a su vez satisfacen las necesidades del usuario con mayor exactitud. El

proceso de integración es continuo, por lo que el esfuerzo final para la integración es nulo.

Al seleccionarse XP como metodología, el proceso de creación del sistema se verá guiado por las siguientes fases (Beck, 1999):

Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto.

Planificación: Se deben realizar las historias de usuarios, con la creación de calendarios, se establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

Diseño: El diseño debe ser simple, con soluciones rápidas para reducir el riesgo. El principal artefacto generado en esta fase son las tarjetas Clase, Responsabilidad y Colaboración.

Codificación: En esta fase se realiza la implementación del sistema guiado por las Tareas de Ingeniería y siguiendo estándares de codificación, esta requiere de pruebas adicionales y revisiones de rendimiento. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual.

Prueba: Todos los códigos sin excepción se les deben realizar pruebas de unidad, y una vez realizado esto pueden ponerse en funcionalidad, cuando un error es encontrado se realiza la corrección, se debe realizar las pruebas con frecuencia y los resultados deben ser publicados.

1.8 Tecnologías y herramientas seleccionadas

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

1.8.1 Tecnologías del lado del cliente

Las operaciones que se realizan en el lado del cliente, generalmente son ejecutadas por mediación de un navegador web. Estas operaciones suelen ser muy rápidas y seguras, puesto que en la mayoría de los casos no es necesario el procesamiento complejo de los datos manipulados. Lo anterior permite reducir la carga de procesamiento que recibe el servidor y la implementación de funcionalidades que doten a las aplicaciones de una mayor usabilidad e interactividad.

HTML 5

HTML, acrónimo de *HyperText Markup Language*, es un lenguaje de publicación especificado como un estándar por el W3C (*World Wide Web Consortium*) que permite la creación de páginas web. HTML es independiente de la plataforma utilizada y se basa fundamentalmente en el uso de etiquetas estructurales y semánticas, adecuadas para la creación de documentos relativamente simples que permiten simplificar su estructura (World Wide Web Consortium, 2017).

CSS 3

Es la última evolución del lenguaje de hojas de estilo en cascada fue utilizado porque presenta muchas novedades como: esquinas redondeadas, sombras, gradientes, transiciones o animaciones, así como nuevos diseños como multi-columnas, cajas flexibles o diseños de rejilla.

JavaScript

Consiste en código que puede ser insertado dentro de un documento HTML, con el objetivo de proporcionarle cierto dinamismo a las páginas web. Es capaz de controlar determinados objetos que componen el entorno en el que se desenvuelve. (Mozilla Project, 2016)

Uno de los aportes más significativos de JavaScript como lenguaje, son todas las librerías que han sido codificadas y que permiten agregarles mayor dinamismo, estética, funcionalidad y usabilidad a los sitios web; entre ellas se destacan: ExtJS, JQuery, Node.JS, entre otras.

1.8.2 Tecnologías del lado del servidor

Las tecnologías en el lado del servidor se encargan de atender las solicitudes de los clientes. Es aquí donde se realiza la mayor parte del procesamiento de la información solicitada, generando una respuesta que puede ser mostrada en formato HTML, XML u otro, que facilite la comprensión por el usuario.

PHP 5.6.25

PHP funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario.

Es seleccionado porque presenta las siguientes características:

- Gratuito: puede descargarse y utilizarse en cualquier aplicación, personal o profesional.
- Eficiencia: puede soportar sin problema millones de visitas diarias.
- Sencilla integración con múltiples bases de datos: Puede conectarse a cualquier sistema gestor de Base de Datos, tanto relacional como no relacional como, por ejemplo: MySQL, PostgreSQL, Oracle, MongoDB, entre otras.
- Versatilidad: PHP puede usarse con la mayoría de sistemas operativos, ya sea basados en UNIX (Linux, Solares, FreeBSD), como con Windows, el sistema operativo de Microsoft.
- Gran número de funciones predefinidas.

JAVA

Es un lenguaje de programación orientado a objetos, específicamente diseñado para tener la menor cantidad de dependencias de implementación como sea posible. Las aplicaciones pueden ejecutarse en cualquier máquina virtual Java (JVM), independientemente de la arquitectura del equipo.

Es seleccionado porque presentan las siguientes ventajas:

- Lenguaje simple: Java no es para nada complejo, su curva de aprendizaje es realmente corta, por lo que de inmediato el usuario podrá familiarizarse con los términos y las funciones que el lenguaje utiliza.
- Lenguaje orientado a objetos: los objetos se encargan de encapsular información, clases y funciones, las cuales se pueden manipular más adelante

o se pueden agregar a distintos programas y existe lo que es la manipulación de datos entre objetos.

- Aplicaciones distribuidas: Java brinda la posibilidad de hacer aplicaciones distribuidas, estas aplicaciones en red, se ejecutan en una plataforma que se compone por una base de cómputo distribuido y funciona perfectamente, lo que propicia que mantenga la estabilidad y el rendimiento se incremente.
- Si se desea ejecutar el programa en otra máquina no es necesario reescribir el código. Un solo código funciona para todos los browsers compatibles con Java o donde se tenga una Máquina Virtual de Java.

1.8.3 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o IDE (siglas en inglés de *Integrated Development Environment*) consiste en un conjunto de herramientas de programación puestas a disposición como un programa informático.

NetBeans

NetBeans IDE le permite desarrollar rápida y fácilmente aplicaciones de escritorio, móviles y web Java, así como aplicaciones HTML5 con HTML, JavaScript y CSS. El IDE también proporciona un gran conjunto de herramientas para desarrolladores de PHP y C / C ++. Es libre y de código abierto y tiene una gran comunidad de usuarios y desarrolladores de todo el mundo.

Es seleccionado porque presentan las siguientes ventajas:

- Edición rápida e inteligente de código.
- Gestión de proyectos fácil y eficiente.
- Desarrollo rápido de la interfaz de usuario.

1.9 Sistema de Gestión de Base de Datos

MySQL 5.7.14

Es un sistema de gestión de bases de datos relacional, desarrollado bajo licencia dual GPL/Licencia comercial por la corporación Oracle. Es destacable, la condición de código abierto, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente.

Según DB-Engines Ranking (DB-Engines, 2017), lista de sistemas de gestión de bases de datos clasificados por su popularidad actual, MySQL tiene el primer puesto dentro de los SGBD de código abierto más preferidos.

MySQL se decide utilizar por los siguientes motivos:

- Velocidad.
- Facilidad de uso.
- Capacidad de gestión de lenguajes de consulta.
- Pueden conectarse muchos clientes simultáneamente al servidor.
- Los clientes pueden utilizar varias bases de datos simultáneamente..
- Dispone de control de acceso.
- Se puede utilizar en una gran cantidad de sistemas Unix diferentes, así como bajo Microsoft Windows.

1.10 Servidor web

Un servidor web, en términos de aplicaciones informáticas, constituye un programa que permite alojar aplicaciones que funcionan comúnmente bajo la filosofía de la arquitectura Cliente – Servidor. Estos sistemas se encargan de procesar las peticiones de los clientes web, para luego devolver una respuesta con los datos procesados, y mostrarlos mediante el uso de un lenguaje de marcas, usualmente HTML.

Apache 2.4.23

Apache 2 es un servidor web de código abierto que implementa el protocolo HTTP 1.1, caracterizado fundamentalmente por su alto nivel de configuración, modularidad, robustez y estabilidad. El mismo, desarrollado bajo la licencia ASF por *The Apache Software Foundation*, es considerado una de las mejores y más aceptadas creaciones del mundo del software libre (NetCraft, 2017).

Teniendo en cuenta las estadísticas históricas y uso diario proporcionadas por NetCraft (2017), este servidor llegó a usarse en 2016 en el 46% de los sitios web en el mundo. Las estadísticas presentadas demuestran que Apache 2 ha sido considerado como el servidor web HTTP por excelencia, logrando que millones de servidores a nivel mundial ratifiquen su utilización.

Apache 2 se decide utilizar por los siguientes motivos:

- Software de código abierto: Además de permitir la alteración y distribución de su código fuente, es un software gratuito, convirtiéndolo en una alternativa de fácil acceso.
- Arquitectura modular: La existencia de módulos con los que Apache se integra posibilita la facilidad en la configuración y ampliación de sus capacidades funcionales por la comunidad de desarrollo.
- Multiplataforma: Puede ser utilizado en una multitud de sistemas operativos, convirtiéndolo en un servidor web prácticamente universal.
- Extensible: Puede aumentar sus capacidades de funcionamiento gracias a las extensiones desarrolladas por la comunidad internacional. Es así como apache ofrece soporte para páginas dinámicas escritas en PHP, Python, Java, Ruby, Perl, entre otros lenguajes de programación.

1.11 Herramienta CASE

Las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Visual Paradigm

Es una herramienta CASE multiplataforma, que soporta el ciclo completo de desarrollo de software: análisis, diseño, implementación y pruebas. Facilita la construcción de aplicaciones informáticas con un menor coste que destacan por su alta calidad y contribuye a mejorar la experiencia de usuario mediante el diseño de un gran número de artefactos de ingeniería de software. Permite la generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería directa e inversa, la gestión de requisitos de software y la modelación de procesos del negocio (Visual Paradigm, 2016).

1.12 Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML) es el lenguaje estándar especificado por el *Object Management Group* (OMG) para visualizar, especificar, construir y documentar los artefactos de un sistema, incluyendo su estructura y diseño. Utiliza un conjunto de símbolos y notaciones para representar gráficamente los diversos componentes que

forman parte de la arquitectura de software. Permite el modelado de procesos de negocio y el modelado de requisitos apoyándose en el análisis orientado a objetos (Object Management Group, 2016).

1.13 Marco de trabajo

Un marco de trabajo o *“framework”*, puede definirse como un conjunto bien estructurado y organizado de bibliotecas que ponen a disposición determinadas funcionalidades básicas implementadas con anterioridad, permitiendo a los programadores centrarse objetivamente en las nuevas funcionalidades a desarrollar.

Symfony 2.8.0

Es un framework diseñado para optimizar el desarrollo de aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Está desarrollado con PHP 5. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows (Potencier, 2017)

Se decide utilizar por los siguientes motivos:

- Tiene por defecto implementados los patrones de diseño situándolos en las capas de Modelo y Controlador que plantea el patrón arquitectónico MVC.
- Código Abierto.
- Utiliza una librería para modelar la base de datos.
- Es posible realizar cambios "en caliente" de la configuración (sin necesidad de reiniciar el servidor).

Doctrine

Doctrine (versión 2.3) es una potente librería para ORM (Mapeo Relacional de Objetos en español) diseñado especialmente para PHP 5.2 o superior y que posee una Capa de Abstracción de Base de Datos (DBAL por sus siglas en inglés).

Entre sus más reconocidas funcionalidades, se destaca la posibilidad de exportación de una base de datos existente a sus clases PHP correspondientes y también a la

inversa, siguiendo las pautas del ORM. Dado que utiliza el patrón Active Record, el mismo permite la manipulación de la base de datos con el uso de la programación orientada a objetos (POO). Gracias a que posee su propio lenguaje de consulta, que consiste en tratar los datos como objetos; se garantiza la independencia de las aplicaciones web hacia los sistemas gestores de bases de datos y un rendimiento óptimo.

1.14 Conclusiones parciales

A partir del análisis del proceso de diagnóstico inicial en los STI estudiados, las entrevistas realizadas a expertos, la selección de las herramientas y tecnologías se arriba a las siguientes conclusiones:

- Se constató que el diagnóstico inicial necesita 2 dimensiones: Clasificación según estilos de aprendizaje y tipo de inteligencia, y agrupamiento conceptual.
- Se definió que la clasificación inicial del estudiante fuera realizada según los estilos de aprendizajes: teórico, visual/verbal y el activo reflexivo, y el tipo de inteligencia lógico-matemática, utilizando como técnica las reglas de producción.
- Se definió que el agrupamiento conceptual se realizará utilizando el algoritmo LC Conceptual.
- Se seleccionó como técnica para la clasificación: las reglas de producción y para el agrupamiento: el algoritmo LC Conceptual.
- 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.
- Las herramientas y lenguajes seleccionados son adecuados para el desarrollo del sistema.

Capítulo 2: Diseño de la propuesta de solución

2.1 Introducción

En este capítulo se abordarán los aspectos fundamentales relacionados con el diseño del sistema a desarrollar. Entre los elementos a destacar se encuentran el diagrama del modelo del dominio, mediante el cual se representan los objetos reales que intervienen en el proceso de diagnóstico inicial. Como vía para definir las futuras funcionalidades se generaron los artefactos relacionados con la metodología utilizada, en las fases de exploración, planificación y diseño.

2.2 Modelo de dominio

La presente propuesta tendrá sus bases en un modelo de dominio⁴, el cual permitirá obtener una mejor comprensión del entorno de la herramienta, mostrando los principales objetos de la vida real con los que trabaja, los cuales forman parte de la propuesta de solución.

Se definen como conceptos fundamentales:

Estudiante: Representa los estudiantes que intervienen en el proceso.

Estilos de aprendizaje: Representa los estilos de aprendizaje que pueden poseer los estudiantes.

Tipo de inteligencia: Representa el tipo de inteligencia que pueden poseer los estudiantes.

STI: Representa el sistema tutorial inteligente con el cual los estudiantes interactúan y realizan el diagnóstico inicial.

Lógica matemática: Representa el tipo de inteligencia que se evalúa en el diagnóstico inicial.

Visual/verbal: Representa uno de los estilos de aprendizaje que se evalúa en el diagnóstico inicial.

Activo Reflexivo: Representa uno de los estilos de aprendizaje que se evalúa en el diagnóstico inicial.

⁴ Modelo de dominio: Representa los objetos significativos en el dominio del problema.

Teórico: Representa uno de los estilos de aprendizaje que se evalúa en el diagnóstico inicial.

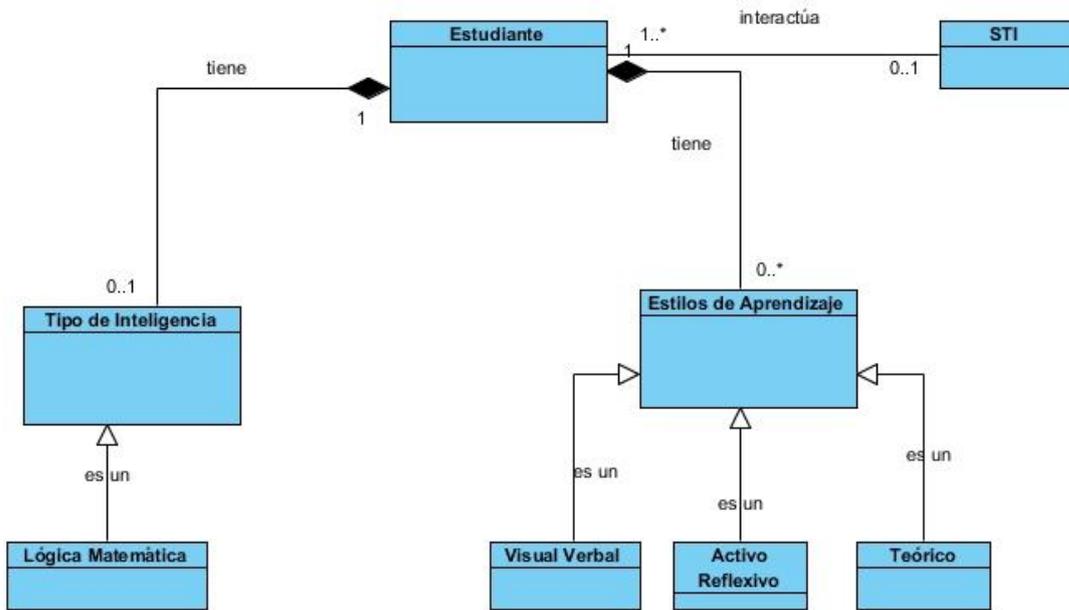


Figura 2 Modelo del dominio

2.3 Propuesta de solución

La herramienta combina las reglas con el agrupamiento conceptual para desarrollar el diagnóstico inicial del estudiante, consta de 2 fases, como se muestra en la figura 3:

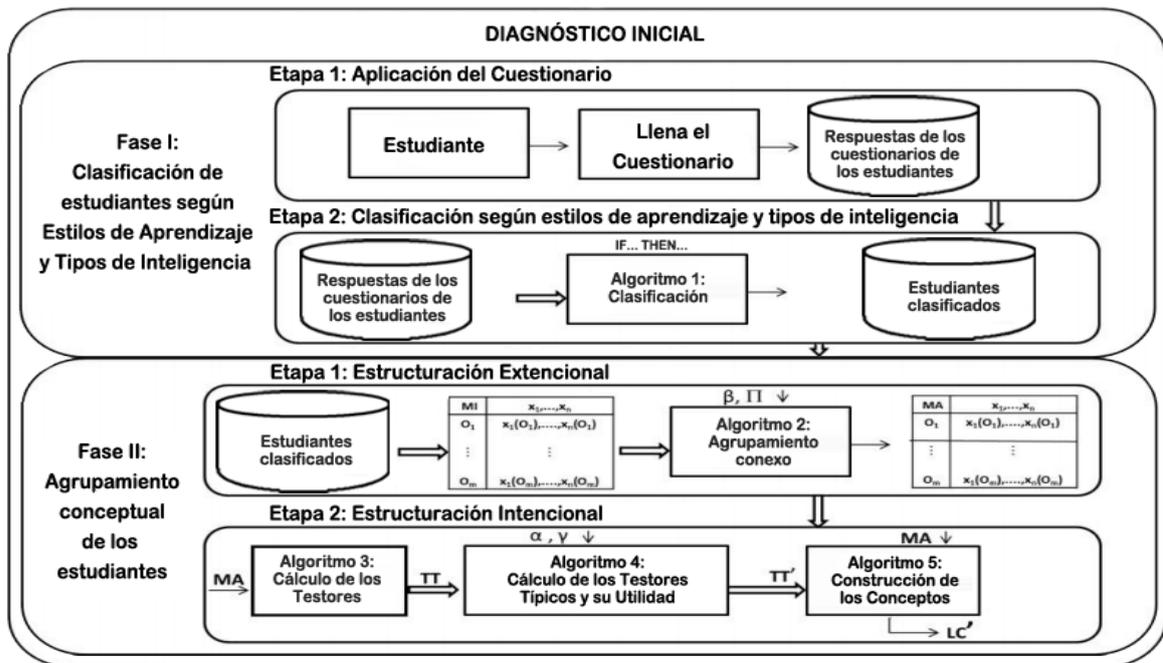


Figura 3 Propuesta de solución (Fase II tomada de Tesis de Maestría de Yunia Reyes)

Fase I. Clasificación de los estudiantes según estilos de aprendizaje y tipo de inteligencia

En esta fase se realiza la clasificación del estudiante a partir de las respuestas de los cuestionarios, se encuentra dividida en dos etapas.

Etapas 1: Aplicación del cuestionario

Esta es la primera etapa de la Fase 1 en la cual los estudiantes responden los cuestionarios y posteriormente es llenada una BD con las respuestas.

Etapas 2: Clasificación de los estudiantes según estilos de aprendizaje y tipo de inteligencia

Algoritmo 1: Clasificación

Entrada: Base de Datos con las respuestas de los cuestionarios.

Salida: Base de Datos con estudiantes clasificados.

Paso 1: Clasificar estudiantes según estilos de aprendizaje.

Ejemplo de reglas utilizada para la clasificación del estilo de aprendizaje visual/verbal:

R1: IF suma ≥ 0 AND suma ≤ 5 ELSE mensaje = " En Ud. no predomina el Canal Visual de percepción de la información".

R2: IF suma ≥ 6 AND suma ≤ 10 ELSE mensaje = " En Ud. predomina el Canal Visual de percepción de la información."

Paso 2: Clasificar estudiantes según tipo de inteligencia.

R1: IF suma ≥ 35 && suma ≤ 50 ELSE mensaje = " Ud. tiene predominio de la inteligencia lógico matemática".

R2: IF suma ≥ 0 AND suma ≤ 34 ELSE mensaje = " No tiene predominio de la inteligencia lógico matemática".

Paso 3: Registrar los resultados en la base de datos.

Fase II. Agrupamiento conceptual de los estudiantes

Esta fase consta de dos etapas, inicialmente se realiza la estructuración extensional en la cual se forman los grupos, luego la estructuración intencional donde se determina el concepto de cada grupo formado en la etapa anterior.

Algoritmo 2: Agrupamiento conexo.

Entrada: Base de Datos con estudiantes clasificados.

Salida: Matriz de Aprendizaje.

Paso 1: Construir la Matriz Inicial a partir de la Base de Datos con estudiantes clasificados.

Paso 2: Construir la Matriz de Semejanza utilizando una función de semejanza β

$$\beta(O_i, O_j) = \sum_{i=1}^n \frac{W_i * C_s(O_i, O_j)}{i}$$

Paso 3: Calcular el umbral de semejanza utilizando el criterio

$$\beta_{0=} \frac{1}{m} \sum_1^m \max\{\beta(I(O_i), I(O_j))\}$$

Paso 4: Agrupar siguiendo el criterio

- $\forall O_i, O_j \in C, O_i \neq O_j, \exists O_{i_1}, \dots, O_{i_q} \in C [O_i = O_{i_1} \wedge O_j = O_{i_q} \wedge \forall p = 1, \dots, q - 1: \Gamma(O_p, O_{p+1}) \geq \beta_0]$.
- $\forall O_i \in \mathbb{O} [O_j \in C \wedge \Gamma(O_i, O_j) \geq \beta_0] \Rightarrow O_i \in C$.
- Todo elemento β_0 -aislado es un núcleo β_0 -conexo (degenerado).

Algoritmo 3: Cálculo de los testores.

Entrada: Matriz de Aprendizaje

Salida: Conjunto de Testores (TT^5)

Paso 1: Calcular la Matriz de Diferencia

Paso 2: Calcular la Matriz Básica

Algoritmo 4: Cálculo de los testores de mayor utilidad.

Entrada: TT

Salida: Conjunto de testores de mayor utilidad (TT')

Paso 1: Calcular el peso ϵ_i de los rasgos X_i que aparecen en la familia de testores.

Paso 2: Seleccionar el conjunto de rasgos relevantes ($\Psi_i(t_i)$).

Paso 3: Seleccionar los p^6 testores de mayor ($\Psi_i(t_i)$).

Algoritmo 5: Construcción de los conceptos.

Entrada: MA, TT'

Salida: Conjunto de los l -complejos para cada grupo $K_i(LC')$.

⁵ **TT:** Las filas de la MB conforman el conjunto de Testores (TT).

⁶ **p:** Parámetro que determina la cantidad de testores de mayor utilidad.

Paso 1: Para cada clase K_i donde $i = 1, \dots, r$ calcular la estrella $G_t (K_i/K_1, \dots, K_{i-1}, K_{i+1}, \dots, K_r)$ que estará formada por los $|TT'|$ l-complejos.

Paso 2: Los conceptos que caracterizan a cada agrupamiento K_i serán los l-complejos obtenidos en el paso anterior.

La herramienta permite obtener un conjunto de clases conformadas por estudiantes con rasgos semejantes, de las cuales se conoce su descripción posibilitando que, al interactuar con el STI, el profesor le asigne un protocolo pedagógico a cada clase y de esta forma facilitar el proceso de enseñanza-aprendizaje.

2.4 Fase de exploración

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

2.4.1 Características funcionales del sistema

En la metodología XP los requerimientos funcionales del sistema se modelan a través de las HU (Historias de Usuario), a continuación, se muestran el nombre de cada una:

HU 1: Autenticar usuario.

HU 2: Registrar datos del usuario.

HU 3: Mostrar cuestionario.

HU 4: Enviar cuestionario.

HU 5: Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia.

HU 6: Listar estudiantes registrados.

HU 7: Filtrar búsqueda.

HU 8: Mostrar respuestas de los cuestionarios.

HU 9: Agrupar estudiantes.

HU 10: Mostrar resultados.

HU 11: Imprimir resultados.

2.4.2 Especificación de las Historia de Usuarios

La especificación de las HU se realiza con el objetivo de determinar cuáles de estas son más vitales resolver y poder realizar una correcta planificación de su implementación.

A continuación, se muestran las historias de usuarios a través de tablas, donde los puntos reales y estimados son representados en semanas.

Tabla 1 HU-5 Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia.

Historia de Usuario	
Número: 5	Título: Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia.
Modificación de historia usuario: Ninguna	
Usuario: estudiante	Iteración: 2
Prioridad: Alta	Puntos Estimados: 3
Riesgo de desarrollo: Alto	Puntos reales: 3
Descripción: Después de que el estudiante haya respondido la encuesta se clasificarán según estilos de aprendizaje y tipo de inteligencia a partir de las reglas definidas por los expertos.	

Tabla 2 HU-9 Agrupar estudiantes.

Historia de Usuario	
Número: 9	Título: Agrupar estudiantes.
Modificación de historia usuario: Ninguna	
Usuario: administrador	Iteración: 3
Prioridad: Alta	Puntos Estimados: 4
Riesgo de desarrollo: Alto	Puntos reales: 4
Descripción: Después de clasificar cada uno de los estudiantes se prosigue a agrupar formando clases en dependencia de las semejanzas entre ellos y después se construyen los conceptos que son las características más representativas de cada clase.	

2.4.3 Características no funcionales del sistema

Son las propiedades o cualidades que el producto debe poseer. Debe pensarse en estas propiedades como las características que hacen que el producto sea atractivo, usable, rápido o confiable. A continuación, se presentan los definidos para la realización de la aplicación a desarrollar.

Requerimientos de diseño e implementación

RNF 1. Se requiere el uso del paradigma de Programación Orientada a Objetos (POO).

RNF 2. Se requiere de MySQL como Sistema gestor de base de datos.

RNF 3. Se requiere el uso de Symfony 2.8 o una versión superior LTS (Long Time Support).

Requerimientos de software

RNF 4. Se requiere que la aplicación pueda funcionar en cualquier sistema operativo.

RF5. Se requiere que en la PC Servidor estén instalados las siguientes herramientas:

- Servidor Apache 2.0 o superior.
- Máquina Virtual de Java.
- MySQL como sistema gestor de base de datos.
- Framework Symfony 2.8 o superior.

Requerimientos de hardware

RNF 6. Se requiere para la PC Servidor se las siguientes prestaciones:

- Procesador: Doble Micro a 1.6 GHz a 2.8 GHz
- RAM: 4GB o superior
- Disco duro: 150 GB o superior
- Tarjeta de Red: 1 Gb

RNF 7. Se requiere para la PC Cliente se necesitan las siguientes prestaciones:

- Procesador: 1.6 GHz a 2.8 GHz
- RAM: 1Gb o 2Gb
- Tarjeta de Red: 1 Gb

Requerimientos de seguridad

RNF 8. Se requiere un mecanismo que permita el control de acceso a la aplicación, para contribuir a la confidencialidad y no divulgación de la información manejada por el sistema.

RNF 9. Se requiere la encriptación de la contraseña de los usuarios del sistema, en el trayecto desde la PC Cliente al Servidor.

Requerimientos de interfaz

RNF 10. Se requiere que los textos y mensajes que aparecen en la herramienta sean en idioma español.

RNF 11. Se requiere que la interfaz de la aplicación a desarrollar deba ser sencilla para facilitar la navegación de los usuarios con la herramienta.

2.5 Fase de planificación

La planificación es una fase en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario y las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas o de Iteraciones.

2.5.1 Plan de iteraciones

En esta fase el cliente establece la prioridad de cada historia de usuario, y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Se propone el siguiente plan de entregas para el cual se tuvo en cuenta la prioridad de las características funcionales:

Tabla 3 Plan de iteraciones

Entregable	Iteración	Fin de la iteración
Autenticar usuario	1	Noviembre 2016
Registrar datos del usuario	1	Noviembre 2016
Mostrar cuestionarios	1	Noviembre 2016
Enviar cuestionarios	1	Noviembre 2016
Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia.	2	Diciembre 2017
Listar estudiantes registrados.	2	Diciembre 2017
Filtrar búsqueda.	2	Enero 2017
Mostrar respuestas de los cuestionarios.	2	Enero 2017
Agrupar estudiantes.	3	Febrero 2017
Mostrar resultados.	3	Marzo 2017
Imprimir resultados	3	Marzo 2017

2.5.2 Plan de entrega

El sistema se realizará en 3 iteraciones, durante las cuales se codificarán incrementalmente las once historias de usuario definidas.

Tabla 4 Plan de entrega

Iteración	Historia de usuario	Semanas estimadas
1	Autenticar usuario Registrar datos del usuario Mostrar cuestionarios Enviar cuestionarios	4
2	Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia. Listar estudiantes registrados. Filtrar búsqueda. Mostrar respuestas de los cuestionarios.	6
3	Agrupar estudiantes. Mostrar resultados. Imprimir resultados.	7

2.6 Fase de diseño del sistema

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, dando soporte a las características funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables y sistemas operativos, que debe poseer la aplicación.

2.6.1 Estilo arquitectónico

En las siguientes figuras se representa la estructura de la herramienta utilizando el patrón Modelo Vista Controlador (MVC).

- **Modelo**

Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. A continuación, se muestran las clases que lo conforman:

Student: Clase encargada de almacenar los datos del usuario.

Survey: Clase encargada de almacenar las preguntas del cuestionario por cada sección.

Data_processed: Clase encargada de almacenar las clasificaciones de los estudiantes.

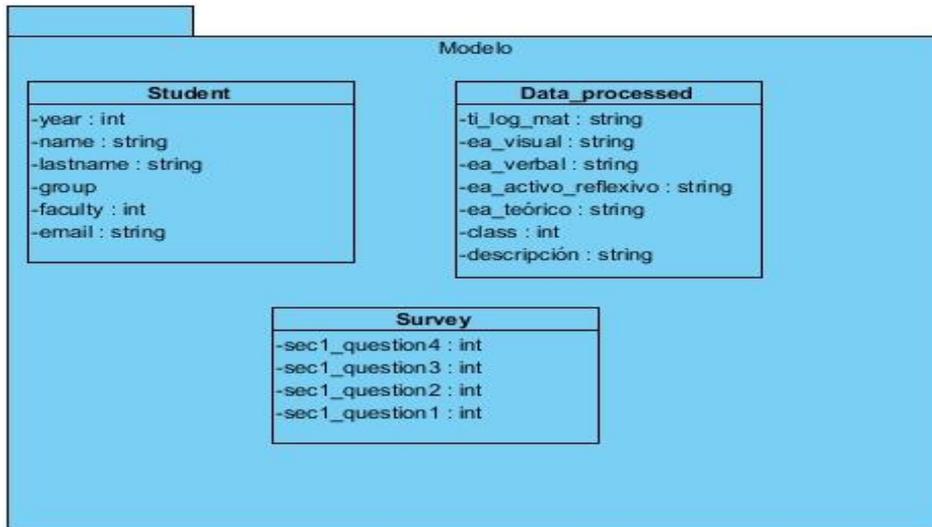


Figura 4 Clases del modelo

- **Vista**

Transforma el modelo en una página web, permite al usuario interactuar con ella.

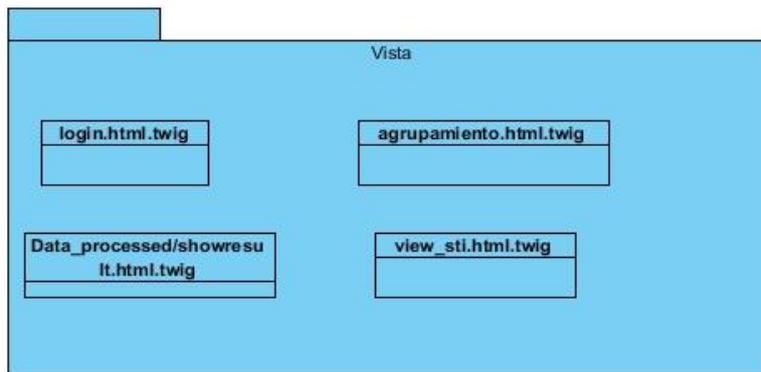


Figura 5 Clases de la vista

- **Controlador**

Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

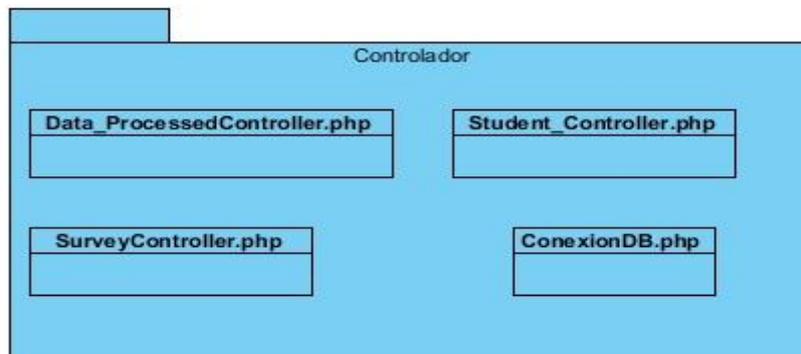


Figura 6 Clases del controlador

A continuación, se muestran las clases que lo conforman:

Data_ProcessedController: se encarga de controlar la gestión de los datos procesados.

Student_Controller: clase encargada de controlar la gestión de los estudiantes.

SurveyController: clase encargada de controlar la gestión del cuestionario.

ConexionDB: clase encargada de establecer la conexión a la base de datos.

2.6.2 Patrones de diseño

En el diseño de la herramienta de clasificación se tuvieron en cuenta los siguientes patrones GRASP (Patrones Generales de Software para Asignación de Responsabilidades), que describen los principios fundamentales de la asignación de responsabilidades a objetos:

Bajo acoplamiento: El acoplamiento es una medida de la fuerza con que un elemento está conectado, tiene conocimiento o confía en otros elementos. El objetivo de este patrón consiste en mantener un bajo nivel de dependencia de otros elementos, por lo que constituye un principio que debe estar presente en todas las decisiones de diseño con lo que se reduce el impacto de los cambios. En Symfony la clase Actions hereda únicamente de sfActions para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Alta cohesión: En el diseño orientado a objetos, la cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento (clase o subsistema). Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que colaboran entre sí y con otros objetos para simplificar su trabajo. Una clase con alta cohesión es relativamente fácil de mantener, entender y reutilizar. En la aplicación existen varios ejemplos, uno de ellos es la clase Action, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, tales como indexAction, showAction y execAction.

Controlador: Este patrón tiene como objetivo asignar la responsabilidad a una clase de recibir o manejar un mensaje de evento del sistema generado por un actor externo, por lo general a través de una interfaz gráfica de usuario a la que accede una persona

para realizar ciertas operaciones en el sistema. En la aplicación se encuentran las siguientes clases como Student_Controller, SurveyController y Data_ProsecedController.

2.6.3 Diseño de la Base de Datos

La herramienta propuesta hace uso de una base de datos, ya que es imprescindible que una vez que el estudiante haya respondido el diagnóstico inicial, las respuestas sean almacenadas para posteriormente se proceda a la clasificación y al agrupamiento.

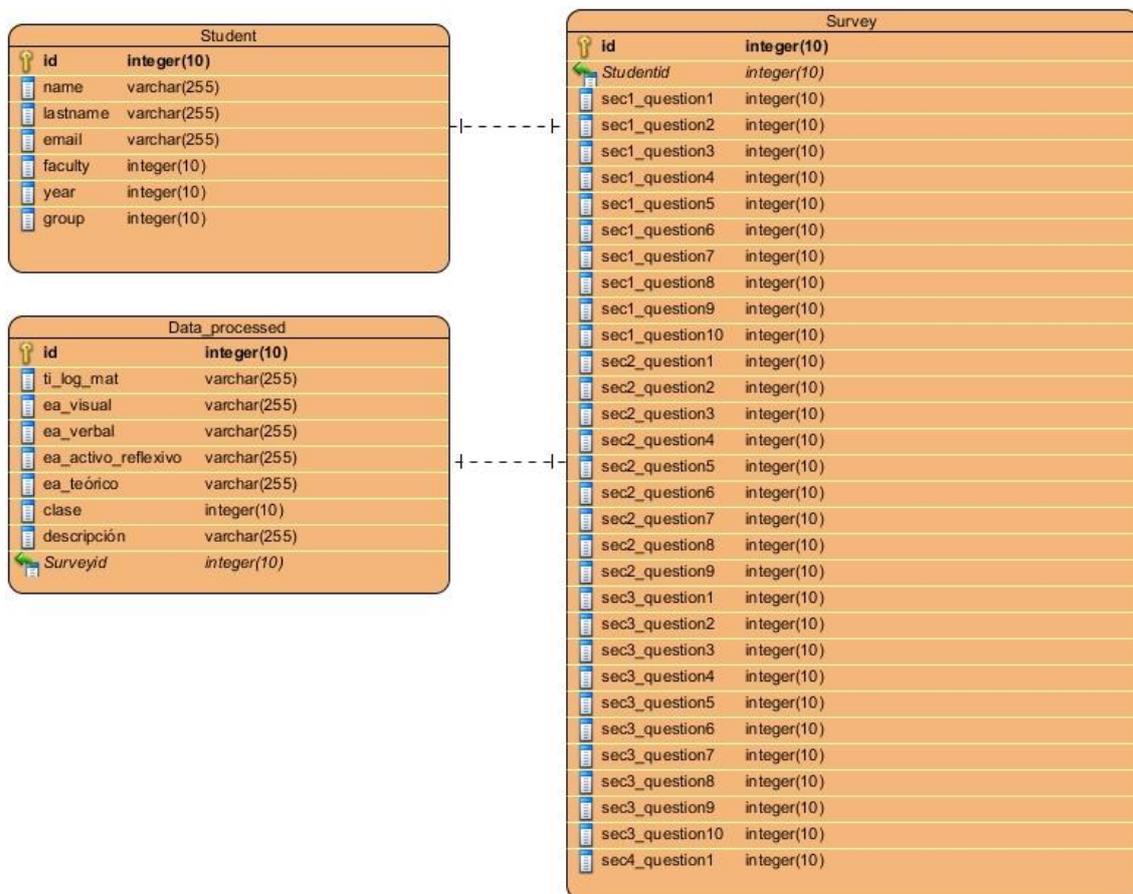


Figura 7 Modelo de la Base de Datos

A continuación, se describen las entidades que conforman la base de datos, que se ilustran en la figura 5:

Student: Almacena la información referente a cada estudiante.

Data_processed: Almacena los resultados de la clasificación según los estilos de aprendizaje y tipo de inteligencia. Algunos de los campos que la componen son: tipo de inteligencia lógica matemática(ti_log_mat), estilo de aprendizaje visual (ea_visual),

estilo de aprendizaje verbal (ea_verbal), estilo de aprendizaje activo-reflexivo (ea_activo_reflexivo), entre otros datos que se pueden apreciar en la figura 7.

Survey: Almacena las respuestas de los cuestionarios dadas por los estudiantes, pueden observar que los campos hacen alusión a cada una de las preguntas de cada sección como, por ejemplo: sec1_question1(pregunta 1 sección 1).

2.6.4 Tarjetas CRC

Las tarjetas CRC (Clase - Responsabilidad - Colaboración), constituyen uno de los artefactos de la metodología XP que guía el proceso de desarrollo de la solución propuesta. Se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores.

Tabla 5 Tarjeta CRC DataProcessed

Clase: DataProcessed	
Responsabilidades:	Colaboradores:
En esta clase es la encargada de realizar el preprocesamiento de los datos y realizar la clasificación inicial.	<ul style="list-style-type: none"> • ConnectionDB

Tabla 6 Tarjeta CRC InitSurvySTI_LC_Conceptual

Clase: InitSurvySTI_LC_Conceptual	
Responsabilidades:	Colaboradores:
Esta clase es la encargada de controlar el proceso de ejecución del algoritmo	<ul style="list-style-type: none"> • DataProcessed • LCConceptual_alg • Testores_alg • ConnectionDB

Tabla 7 Tarjeta CRC LCConceptual_alg

Clase: LCConceptual_alg	
Responsabilidades:	Colaboradores:
Esta clase es la encargada de llamar la conexión a la BD y ejecutar el algoritmo de agrupación.	<ul style="list-style-type: none"> • LCConceptual • ConnectionDB

Tabla 8 Tarjeta CRC MatrizBinaria

Clase: MatrizBinaria	
Responsabilidades:	Colaboradores:
En esta clase se encuentra implementado el cálculo de testores típicos generalizado	Testores_alg

Tabla 9 Tarjeta CRC Testores_alg

Clase: Testores_alg	
Responsabilidades:	Colaboradores:
Esta clase es la encargada de calcular los testores candidatos a ser procesados como testores típicos para determinar los conceptos	<ul style="list-style-type: none"> • ConnectionDB

Tabla 10 Tarjeta CRC FastBR

Clase: FastBR	
Responsabilidades:	Colaboradores:
Esta clase es la encargada de calcular los testores típicos los cuales son usados para determinar los conceptos	<ul style="list-style-type: none"> • ConnectionDB • Testores_alg

Tabla 11 Tarjeta CRC ConnectionDB

Clase: ConnectionDB	
Responsabilidades:	Colaboradores:
En esta clase es la encargada de abrir y cerrar las conexiones a la BD así como ejecutar las consultas SQL	

2.7 Conclusiones parciales

Con el desarrollo de las fases de: exploración, planificación y diseño propuestos por la metodología XP:

- Se determinaron las características funcionales representadas en 11 historias de usuario, planificándose tres iteraciones que permiten organizar el desarrollo del sistema.
- La utilización de patrones de diseños y arquitectónicos, permitió una adecuada organización de los elementos definidos en la propuesta de solución para el desarrollo del sistema.

- Se estimó que el software estará listo en aproximadamente 5 meses.

Capítulo 3: Implementación y Prueba de la herramienta.

3.1 Introducción

Dentro del proceso de desarrollo de software la etapa de implementación, es el proceso de convertir una especificación del sistema en un sistema ejecutable. La misma implica los procesos de transformar las clases y objetos en ficheros fuente, binarios y ejecutables. En la etapa de pruebas el resultado final, o sea el ejecutable, es evaluado en cuanto a su calidad y desempeño como producto de software. En el capítulo se detallan estas fases.

3.2 Implementación

Para la implementación de las historias de usuarios es necesario crear un conjunto de tareas de ingeniería (TI) que faciliten el trabajo de los programadores. Las tareas de realizarse en cada una de las iteraciones se especifican en la próxima tabla:

Tabla 12 Tareas de ingeniería

Iteración	Historia de usuario	Tareas de la ingeniería
1	Autenticar usuario	TI 1. Autenticar usuario
	Registrar datos del usuario	TI 2. Registrar datos del usuario
	Mostrar cuestionarios	TI 3. Mostrar cuestionarios
	Enviar cuestionarios	TI 4. Enviar cuestionarios
2	Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia.	TI 5. Analizar cuestionario. TI 6. Clasificar estudiante.
	Listar estudiantes registrados.	TI 7. Listar estudiantes registrados.
	Filtrar búsqueda.	TI 8. Filtrar búsqueda.
	Mostrar respuestas de los cuestionarios.	TI 9. Mostrar respuestas de los cuestionarios.
3	Agrupar estudiantes.	TI 10. Cálculo de la Matriz de Diferencia TI 11. Cálculo de la Matriz Básica TI 11. Selección de los testores TI 13. Calcular relevancia de cada rasgo TI 14. Selección de los testores más relevantes. TI 15. Generación de los I-complejos de cada grupo.
	Mostrar resultados	TI 16. Mostrar resultados.
	Imprimir resultados	TI 17. Imprimir resultados

Las TI implementadas y definidas en cada iteración serán representadas mediante una tabla, a continuación, se presenta las TI correspondientes a la HU Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia. Para consultar el resto de las tareas [ver Anexo 3.](#)

Tabla 13 Tarea de ingeniería: Analizar cuestionario.

Tarea	
Número de la tarea: 5	Número de la historia: 5
Nombre de la tarea: Analizar cuestionario.	
Tipo de tarea: Implementación	Estimación: 15 días
Programador responsable: Beatriz Lugo - Liomar Rodríguez	
Descripción: Implementar las clases para analizar el cuestionario.	

Tabla 14 Tarea de ingeniería: Clasificar estudiante .

Tarea	
Número de la tarea: 6	Número de la historia: 6
Nombre de la tarea: Clasificar estudiante.	
Tipo de tarea: Implementación	Estimación: 15 días
Programador responsable: Beatriz Lugo - Liomar Rodríguez	
Descripción: Implementar las clases para clasificar utilizando reglas.	

3.2.1 Diagrama de despliegue

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para describir los principales nodos físicos que se necesitan para desplegar el sistema y la relación entre ellos mediante los protocolos de comunicación. En la figura 8, se muestra una propuesta de la vista de despliegue de la aplicación.

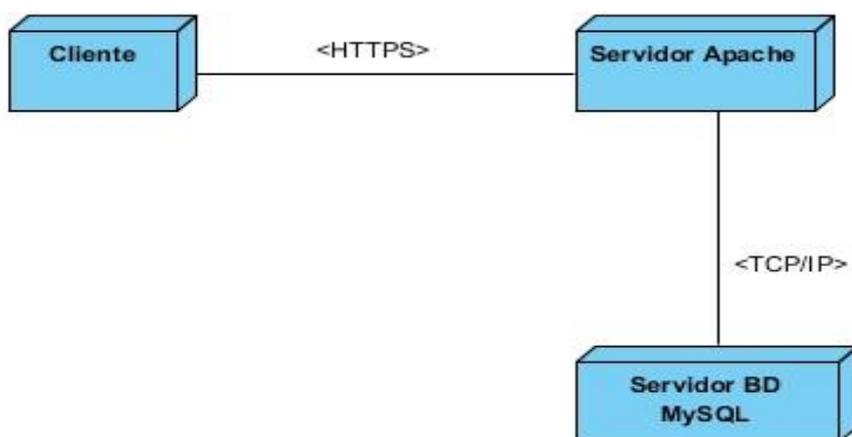


Figura 8 Diagrama de despliegue

A continuación, se expone una breve descripción de los nodos y sus conexiones:

Cliente: Este nodo representa todos los usuarios que se pueden conectar a la aplicación utilizando https.

Servidor Apache: Este nodo representa el servidor donde se encuentra la aplicación. Se conecta con el Servidor BD a través del protocolo TCP/IP.

Servidor BD: Este nodo representa la base de datos donde se guarda las respuestas del cuestionario y además se almacenan los resultados de las clasificaciones y agrupaciones.

3.2.2 Estándares de codificación

Un estándar de codificación es una tecnología, formato o método desarrollado, adoptado a través de proceso abierto de consenso, con la ventaja de facilitar que el código fuente de cualquier aplicación sea legible, se le pueda dar mantenimiento, sea interoperable y pueda ser distribuido.

En la propuesta de solución para declarar el nombre de las variables, métodos y clases se tendrán en cuenta las siguientes convenciones:

- En todos los casos se utilizarán nombres descriptivos que ayuden a una mejor comprensión del código.
- No se utilizará una misma línea para definir más de una variable y siempre que sea posible éstas se inicializarán en su misma línea de declaración.
- Las clases deben comenzar con mayúscula y en caso de estar conformada por palabras compuestas, están unidas y cada palabra debe iniciar con mayúscula siguiendo el estilo determinado.
- Los métodos deben comenzar con letra mayúscula, escribiéndose de manera seguida. En caso de estar conformados por palabras compuestas, a partir de la segunda palabra debe comenzar con letra mayúscula.
- Las variables comienzan con minúsculas, aquellas que sean compuestas están unidas, comenzando minúscula la segunda palabra.

3.3 Pruebas

La prueba del software es un elemento crítico para la garantía de su calidad y representa una revisión final de las especificaciones del diseño y de la codificación. A la herramienta implementada se le aplicó las pruebas unitarias y de aceptación, las cuales propone XP, además fue necesario realizar pruebas de seguridad y de usabilidad, con el objetivo de lograr que el usuario interactúe con el sistema de forma fácil y que la información que se almacene sea accesible para las personas correctas.

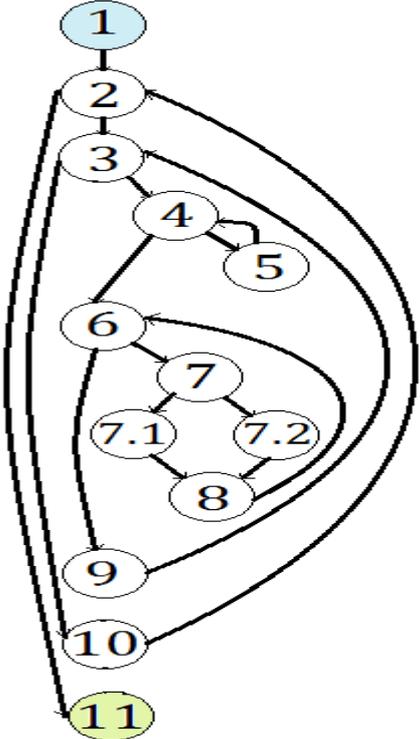
3.3.1 Pruebas unitarias

Para realizar esta prueba se utilizó la técnica del camino básico, estas garantizan que se ejecute cada instrucción del programa por lo menos una vez durante la prueba.

La complejidad ciclomática se basa en la teoría gráfica y se calcula de tres maneras:

1. Número de regiones
2. Complejidad ciclomática es igual a número de aristas, menos el número de nodos más 2 $V(G) = A - N + 2$
3. Complejidad ciclomática es igual al número de nodos predicado más uno
4. $V(G) = P + 1$

A continuación, se muestra la ejecución de la técnica al código implementado para la creación de la Matriz de Aprendizaje.

Camino Básico	Cálculo de la complejidad ciclomática
	<ol style="list-style-type: none"> 1. $V(G) =$ Cantidad de Regiones $V(G) = 6$ 2. $V(G) =$ Aristas - Nudo + 2 $V(G) = 17 - 13 + 2$ $V(G) = 6$ 3. $V(G) = P + 1$ $V(G) = 5 + 1$ $V(G) = 6$

A partir de esta complejidad ciclomática podemos obtener la cantidad de caminos independientes, en este caso son 6 y son los siguientes:

Camino 1: 1-2-3

Camino 2: 1-2-3-10-2-11

Camino 3: 1-2-3-4-6-7-7.1-8-6-9-3-10-2-11

Camino 4: 1-2-3-4-6-7-7.2-8-6-9-3-10-2-11

Camino 5: 1-2-3-4-5-4-6-7-7.1-9-3-10-2-11

Camino 6: 1-2-3-4-5-4-6-7-7.2-9-3-10-2-11

Después de identificar los caminos independientes se preparan los casos de prueba que obliguen la ejecución de cada camino del conjunto básico:

Camino 1

- Entrada: Arreglo bidimensional con los resultados de la clasificación.
- Salida: Arreglo bidimensional vacío.
- Precondiciones: ---

Camino 2

- Entrada: Arreglo bidimensional con los resultados de la clasificación.
- Salida: Arreglo bidimensional vacío.
- Precondiciones: ---

Camino 3

- Entrada: Arreglo bidimensional con los resultados de la clasificación.
- Salida: Arreglo bidimensional de tamaño $n \times n$ dependiendo de la cantidad de filas del arreglo con los resultados de la clasificación, con valores de cada comparación.
- Precondiciones: ---

Camino 4

- Entrada: Arreglo bidimensional con los resultados de la clasificación.
- Salida: Arreglo bidimensional de tamaño $n \times n$ dependiendo de la cantidad de filas del arreglo con los resultados de la clasificación, con valores de cada comparación.
- Precondiciones: ---

Camino 5

Entrada: Arreglo bidimensional con los resultados de la clasificación.

- Salida: Arreglo bidimensional de tamaño $n \times n$ dependiendo de la cantidad de filas del arreglo con los resultados de la clasificación, con valores de cada comparación.
- Precondiciones: ---

Camino 6

Entrada: Arreglo bidimensional con los resultados de la clasificación.

- Salida: Arreglo bidimensional de tamaño $n*n$ dependiendo de la cantidad de filas del arreglo con los resultados de la clasificación, con valores de cada comparación.
- Precondiciones: ---

3.3.2 Pruebas de aceptación

Las pruebas de aceptación en XP son especificadas por el cliente, se centran en las características y funcionalidades generales del sistema. Se definen para cada historia de usuario y los clientes son responsables de verificar que los resultados de estas pruebas sean correctos.

A continuación, se detallan las prueba de aceptación correspondiente a la funcionalidad de Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia, para consultar los restantes casos de pruebas [ver Anexo 4](#).

Tabla 15 Caso de Prueba de Aceptación 9

Caso de Prueba de Aceptación	
Código: P9	Número de la historia:5
Nombre: Clasificar estudiante según estilos de aprendizaje y tipo de inteligencia.	
Descripción: Con esta prueba se verifica que los estudiantes son clasificados .	
Condiciones de ejecución: Debe estar corriendo el servidor Apache.	
Entradas/Pasos de ejecución: 1- Ejecutar la aplicación. 2- Responder cuestionarios. 3- Enviar cuestionarios.	
Resultado esperado: Se espera que las clasificaciones se almacenen correctamente en la BD.	
Evaluación de la prueba: Prueba satisfactoria	

Resultados de las pruebas de aceptación.

Para llevar a cabo la detección de las no conformidades presentes en la aplicación desarrollada, se realizaron 3 iteraciones de pruebas funcionales. En la siguiente tabla se muestran los resultados obtenidos.

Tabla 16 Cantidad de no conformidades por cada iteración de las pruebas de aceptación.

No conformidades	Iteración 1	Iteración 2	Iteración 3
Detectadas	4	2	0
Resueltas	3	2	0
Pendientes	1	0	0

En la siguiente figura se puede apreciar el comportamiento de las no conformidades detectadas en cada iteración.

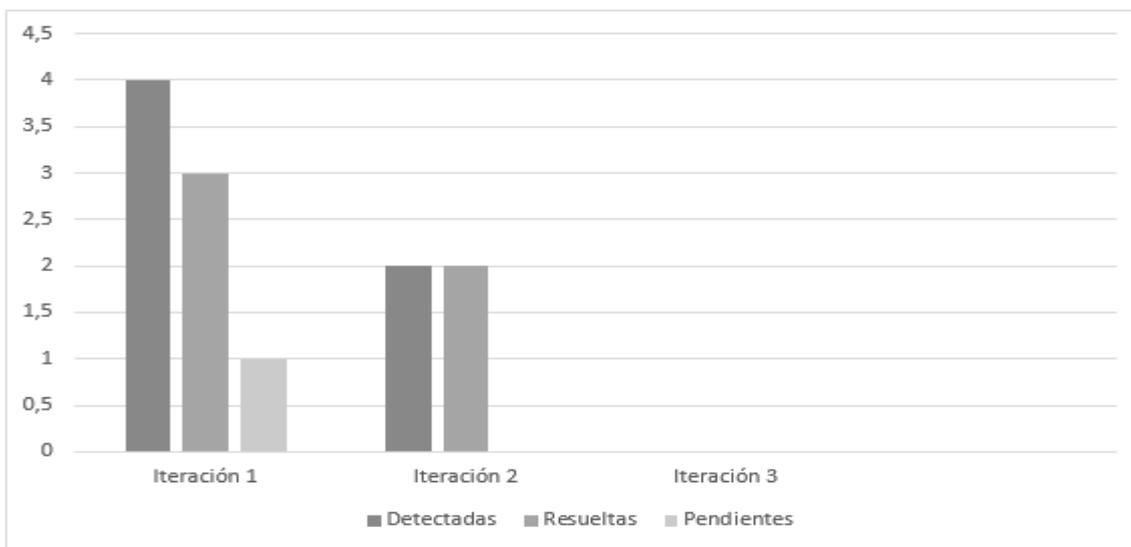


Figura 9 Comportamiento de las no conformidades por cada iteración de las pruebas de aceptación.

Las no conformidades detectadas pueden ser agrupadas atendiendo a diferentes tipologías, en este caso se detectaron 4 de tipo mensajes incorrectos en la aplicación y 2 de errores de validación de campos.

1.3.3 Pruebas de usabilidad

La realización de pruebas de usabilidad contribuye, en cierta medida, a la adquisición de aplicaciones de alta calidad y gran facilidad de uso por parte de los usuarios finales. Para ello, especialistas del centro de calidad de la UCI, han establecido una lista de chequeo que proporciona un conjunto de preguntas clasificadas en 9 categorías. A continuación, se muestran los resultados de la aplicación de la lista de chequeo a la herramienta informática para el diagnóstico inicial en la figura 8 se visualiza el cumplimiento de los indicadores.

Tabla 17 Cumplimiento de la lista de chequeo de usabilidad.

Categorías de Indicadores	Indicadores	Correctos
Visibilidad del sistema	17	17
Lenguaje común entre sistema y usuario	11	9
Libertad y control por parte del usuario	30	25
Consistencia y estándares	33	30
Estética y diseño minimalista	16	10
Prevención de errores	8	8
Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	12	12
Ayuda y documentación	11	11
Flexibilidad y eficiencia de uso	6	6
TOTAL	144	128

Se puede apreciar que, de un total de 144 indicadores de usabilidad, la herramienta implementada cumple con 128 indicadores, cifra que representa el 88,9 %.

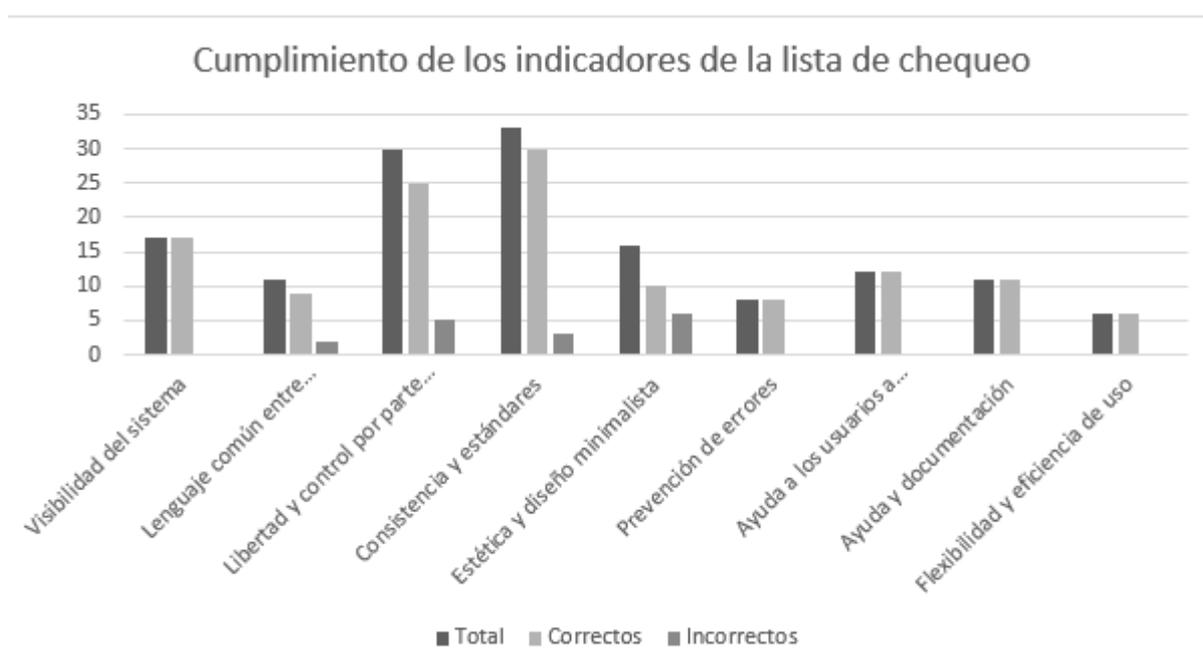


Figura 10 Cumplimiento de los indicadores de la lista de chequeo

Después de analizar los resultados obtenidos en las pruebas de usabilidad, se identificaron 4 indicadores con posibles mejoras de acuerdo al alcance del presente trabajo, los cuales son: lenguaje común entre sistema y usuario, libertad y control por parte del usuario, consistencia y estándares, y por último estética y diseño minimalista. Estos fueron corregidos parcialmente en función de elevar el nivel de facilidad de uso e interacción entre el usuario y las operaciones que se realizan en la herramienta;

incrementando el nivel de usabilidad hasta el 95,8%. En la gráfica siguiente se representa el estado del nivel de usabilidad resultante.

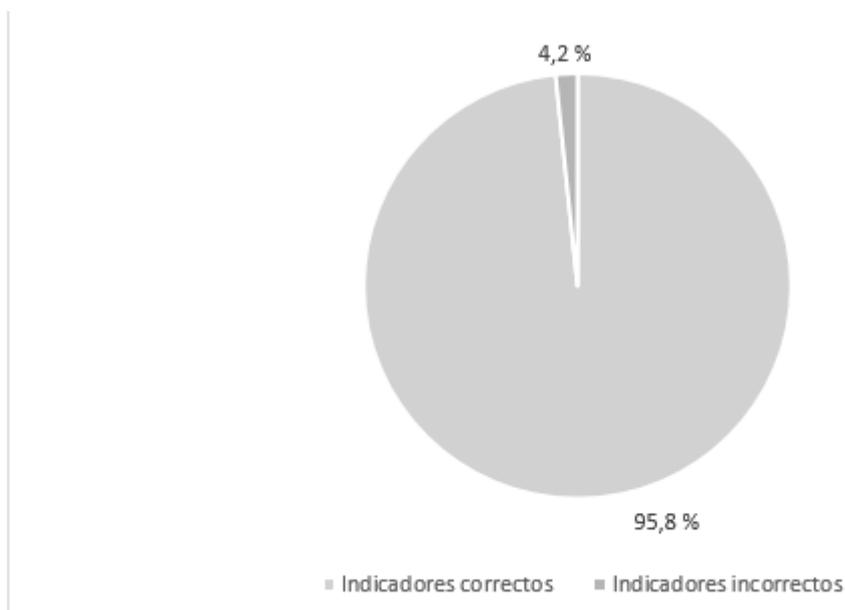


Figura 11 Resultado final de la prueba de usabilidad

De acuerdo a la usabilidad resultante, se puede constatar que la herramienta para el diagnóstico inicial de los estudiantes en el STI favorece el proceso de aprendizaje, contribuyendo de forma fácil y sencilla la interacción de los usuarios con ella.

1.3.4 Pruebas de seguridad

La realización de pruebas de seguridad contribuye a la detección temprana de vulnerabilidades y la toma de medidas para la disminución de amenazas de ataque, y con ello de cómputo más seguros y confiables. En este sentido, los especialistas del centro de calidad de la UCI definieron una lista de chequeo que establece 14 indicadores categorizados en 4 tipos de pruebas.

Después de aplicar la lista de chequeo a la herramienta, se obtuvieron los siguientes resultados:

Tabla 18 Resultados de la aplicación de la lista de chequeo para pruebas de seguridad .

Tipo de prueba	Indicadores	Correctos
Pruebas de autorización	2	2
Pruebas de gestión de sesiones	2	2
Comprobación del sistema de autenticación	6	3
Validación de datos	4	3

Los indicadores de la lista de chequeo aplicada evaluados como incorrectos, fueron corregidos oportunamente según se describe en cada indicador; proporcionando de esta forma mayor seguridad a la herramienta de diagnóstico inicial.

3.4 Conclusiones parciales

Luego de implementar la propuesta de solución y de haber realizado las pruebas se arriba a las siguientes conclusiones:

- Las pruebas unitarias realizadas demostraron que no existían errores de implementación.
- A través de las pruebas de aceptación se corrigieron las 6 no conformidades encontradas: 4 de tipo mensajes incorrectos en la aplicación y 2 de errores de validación de campos.
- Mediante las pruebas de usabilidad se aseguró que la interacción de los usuarios con la herramienta fuese fácil y sencilla.
- A través de las pruebas de seguridad se confirmó que la herramienta cumplía con los requisitos de este tipo establecidos.

CONCLUSIONES GENERALES

En el presente trabajo se ha llevado a cabo un proceso de desarrollo de software completo, dividido en flujos de trabajos e iteraciones, con el objetivo de lograr un producto de calidad en el tiempo establecido. Una vez completada la presente investigación, se puede concluir que:

- Se obtuvo una herramienta informática para realizar el diagnóstico inicial del estudiante en el Sistema Tutorial Inteligente para Reconocimiento de Patrones.
- El diagnóstico inicial está conformado por dos dimensiones: una primera que realiza la clasificación inicial de los estudiantes identificando los estilos de aprendizajes y el grado que presentan de la inteligencia lógico matemática, y en la segunda mediante agrupamiento conceptual se obtienen los grupos de estudiantes semejantes con su respectiva descripción.
- La selección de XP como metodología de software, MySQL como gestor de base de datos, Symfony como framework de desarrollo y Apache como servidor web, permitió el desarrollo adecuado de la herramienta.
- La evaluación de las pruebas de software garantizó erradicar las insuficiencias detectadas en la herramienta.

RECOMENDACIONES

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

- Tener en cuenta para el diagnóstico inicial las características propuestas por Zulma Cataldi: competencias, conocimientos básicos de la asignatura, conocimientos previos, rasgos afectivos, emocionales y psicológicos.
- Implementar nuevas funcionalidades que permita al experto crear cuestionarios para medir las características mencionadas anteriormente sin necesidad de programar.
- Realizar la integración de la herramienta de diagnóstico inicial con el STI para la asignatura de Reconocimiento de Patrones.

REFERENCIAS BIBLIOGRÁFICAS

Krusche, H. 1996. La rana sobre la mantequilla: PNL :Fundamentos de la programación neurolingüística .Alemania : Editorial sirio. SA 200684-7808--186-0.

Zafra. 2013. Aprendizaje de múltiples instancias para la clasificación de estudiantes en sistemas de gestión del aprendizaje. España

Argueta Antonio Nolasco . 2012.

Cataldi Zulma. 2012. El problema del modelado del estudiante en Sistemas Tutores Inteligentes. Argentina.

Ferreiro, Salcedo. 2012.ELE-Tutor Sistema Tutorial para el español como Lengua Extranjera. Chile .

Ferreira. 2015. ELE-Tutora: Un Sistema Tutorial Inteligente para la focalización de los errores gramaticales en la enseñanza del español como Lengua Extranjera.

UNESCO. 2013. Enfoques estratégicos sobre las TICS en educación en América Latina. Santiago de Chile : Oficina Regional de Educación de América Latina.

Agudelo. 2010. Estilos de aprendizaje basados en el modelo de Kolb en la educación virtual. Guadalajara : Revista de innovación educativa.

Ospina. 2010. Estilos de aprendizaje y rendimiento académico en estudiantes de Psicología. Argentina .

Molinari. 2013. Herramienta para la clasificación de estudiantes basados en la lógica difusa para evaluar planes de estudio individual. Rusia.

Araujo. 2015.Herramienta para mejorar las recomendaciones Pedagógicas mediante la clasificación a los estudiantes de acuerdo a su Comportamiento Interaccional en un Ambiente de Aprendizaje Gamificado. Brasil .

Gardner, Howard. 1998. Inteligencias Múltiples.

Martínez, Raidell Avello. 2014. Los Entornos Virtuales de Aprendizaje como recurso didáctico en el ámbito universitario. 1, Cienfuegos : Revista Universidad y Sociedad, 2, 卷 5.

Reynoso. 2004. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. [online] [Citado el: 20 de octubre de 2016.] [http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/style.asp].

Pérez y otros.2012. Sistema Inteligente para el Algebra Lineal. Camagüey.

Morales. 2012. Sistema Tutorial Inteligente de apoyo al proceso de enseñanza-aprendizaje de la asignatura de Sistemas Operativos. La Habana.

Strasser. 2000.Sistema Tutorial Inteligente para el Apoyo de la Enseñanza de la Lectura Inicial.1, Chile : PSYHKE, 2000.

Psicotext. 2014.TuinLec, tutor inteligente para mejorar la competencia lectora. España.

Zatarain. 2014.Tutor Inteligente con reconocimiento y manejo de emociones para Matemáticas. México .

Ortega. 2010.Tutor Virtual para el aprendizaje Autónomo de Idiomas. La Habana.

Apache. [En línea]. Apache Software Foundation, 2016. [Citado el: 4 de Diciembre de 2016]. Disponible en: [<http://www.apache.org>].

Lage Fernando, Zulma Cataldi. El problema del modelado del estudiante en Sistemas Tutores Inteligentes. Argentina: s.n.

ELE-Intelligent Tutor: A computational parser for the processing of grammatical errors in Spanish as a Foreign Language. Santiago de Chile: Register Signos versión On-line. 0718-0934.

Gustavo Pérez, Pablo Ospina. 2010. Engineering Instruction with Intelligent Tutorial Systems. Buenos Aires-Argentina : s.n.

Symfony. [En línea] [Consultado el: 10 de Diciembre de 2016.] Disponible en: [<http://symfony.com>.]

MySQL. [En línea] [Consultado el: 10 de Enero de 2017.] Disponible en: [<https://www.mysql.com/>]

NETCRAFT. Servidores web más usados desde 2000. [En línea]. Sitio web de NetCraft, 2016. [Consultado el: 25 de Diciembre de 2016]. Disponible en: [<http://www.netcraft.com>].

DB-Engines. Ranking de base datos más populares desde 2013. [En línea]. Sitio web de **DB-Engines**, 2017. [Consultado el: 25 de Mayo de 2017]. Disponible en: [<http://www.netcraft.com>].

Beck, Kent.1999. Extreme Programming Explained: Embracing Change.

WORLD WIDE WEB CONSORTIUM (W3C). *What is CSS?* [En línea]. Consultado el: [13 de diciembre de 2017] Disponible en: [<http://www.w3.org/Style/CSS/>].

VISUAL PARADIGM. Visual Paradigm for UML - Software design tools for agile software development. [En línea]. 2016. Consultado el: [22 de octubre de 2016] Disponible en: [<http://www.visual-paradigm.com/product/vpuml/>].

OBJECT MANAGEMENT GROUP. *Introduction to OMG's Unified Modeling Language.* [En línea]. 2013. Consultado el: [25 de octubre de 2016]. Disponible en: [http://www.omg.org/gettingstarted/what_is_uml.htm].

MOZILLA PROJECT. *JavaScript Overview.* [En línea]. Mozilla Developer Network, 2016. [Consultado el: 18 de Diciembre de 2016]. Disponible en: [<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Overview>].

Potencier, Fabian. Symfony2. [Consultado el: 30 de Febrero de 2017]. Disponible en: [<https://Symfony.com>].

BIBLIOGRAFÍA

Colectivo de autores: CUJAE, Folleto Introducción a la Inteligencia Artificial.

Ander-Egg, E. 2006. Teoría del aprendizaje social Inteligencias Múltiples.. Argentina : Homo Sapiens Ediciones.

V, Carlos A. Cuevas. 2001.Una propuesta para la construcción de sistemas tutoriales inteligentes para apoyar la enseñanza de las matemáticas. México : s.n., 2001, Vol. 13.

Zulma Cataldi y Fernando Lage. Sistemas Inteligentes aplicados al modelado del estudiante. [En línea]: Disponible en:[<http://www2.istec.org/wp-content/gallery/ebooks/sibd/docs/sibd09-final9.pdf>.]

Shulcloper, José R. 1997 .Algoritmo LC Conceptual para el agrupamiento de objetos. México : s.n., 1997.

Shulcloper, José R. 2009. Reconocimiento lógico combinatorio de patrones: teoría y aplicaciones. La Habana. Cuba: s.n.

Shulcloper, José R. 1999. LC Conceptual Algorithm: Characterization using typical testors by class.

N., D' Andrea y S., Rodríguez. Sistema tutorial inteligente como espacio didáctico para el aprendizaje en la Universidad Panamericana del Puerto. Venezuela : s.n.

Loinaz, Maite Urretavizcaya. 2001. Sistemas Inteligentes en el ámbito de la Educación. s.l. : Revista Iberoamericana de Inteligencia Artificial., Vol. 12. 1137-3601.

Huapaya, Constanza Raquel. 2009. Sistemas Tutoriales Inteligentes. Un análisis crítico.. Argentina : s.n.

Cataldi, Zulma, Salgueiro, Fernando. Sistemas Tutoriales Inteligentes: los estilos de estudiantes para la selección del tutorizado. Argentina : s.n.

Cataldi, Zulma, Salgueiro, Fernando y otros. Sistemas Tutoriales Inteligentes con modelación del tutor y el estudiante para mejorar los aprendizajes de la programación en ingeniería.. Buenos Aires Argentina : s.n.

González, Yunia R., Sánchez ,Natalia M., 2014. La toma de decisiones en los Sistemas Tutoriales Inteligentes utilizando el agrupamiento conceptual. La Habana, Cuba : Revista Cubana de Ciencias Informáticas, Vol. 8. 2227-1899.

Shulcloper, José R. 2000. Logical Combinatorial Pattern Recognition.. Barcelona, España : s.n.

Los estilos de aprendizaje y la enseñanza de las matemáticas. Madrid, España : Revista Complutense de Educación.

González, Yunia R. Los Sistemas Tutores Inteligentes aplicados en la Educación. La Habana, Cuba : Revista informática-tecnológica. 1995-9419.

Hernández ,Jairo A., Rengifo, Yois Smith. 2015. Los sistemas tutoriales inteligentes y su aplicabilidad en la educación.. Colombia : Revista de la Facultad de Educación, Ciencias Humanas y Sociales, 2015.

Cataldi, Zulma, Lage, Fernando. Metodología extendida para la creación de software educativo desde una visión integradora. Buenos Aires, Argentina : Revista Latinoamericana de Tecnología Educativa, Vol. 2.

Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). España : s.n.

Maraza, Benjamín. 2012. Modelado del estudiante a partir de los estilos de aprendizajes. Perú : Nuevas Ideas en Informática Educativa.

Cataldi, Zulma, Lage, Fernando. Modelado del Estudiante en los Sistemas Tutoriales Inteligentes. Argentina : Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología.

Modelo de Sistema Basado en Casos utilizando el agrupamiento conceptual incremental. La Habana : Serie Científica de la Universidad de las Ciencias Informáticas.

Estilos de aprendizaje y rendimiento académico en estudiantes de Psicología. s.l. : Diversitas: Perspectivas en Psicología, 2010, Vol. 6. 1794-9998.

Baños, Liyanis Peláez. 2015. Sistema Inteligente para el apoyo a la Toma de Decisiones combinando el Razonamiento Basado en Casos y el algoritmo de agrupamiento LC-Conceptual. La Habana, Cuba : s.n., 2015.

Coello, Saida. 2011. . El proceso de investigación científica. La Habana, Cuba : Editorial Universitaria, 2011. . ISBN: 978-959-16-1307-3.

Combinatorial logic conceptual clustering: an alternative to Decision Making. La Habana : Herramientas para la Estructuración Conceptual de Espacios.

Diagnóstico adaptativo del estudiantes en los Sistemas Tutoriales Inteligentes. Chile : Revista Signos, Vol. Estudios de Lingüística. 0718-0934.

Díaz, Darel Camps. 2007. Implementación de algoritmos. La Habana, Cuba : s.n., 2007.

Heidy Fontalvo, Fernando Iriarte,. 2010. Diseño de ambientes virtuales de enseñanza aprendizaje y sistemas hipermedia adaptativos basados en modelos de estilos de aprendizaje .. Barranquilla, Colombia : EDICIONES UNINORTE, 2010, Vol. 8.

Lage Fernando, Zulma Cataldi. El problema del modelado del estudiante en Sistemas Tutores Inteligentes. Argentina: s.n.

Boeras, Mairelys. 2012. Aplicando el método de Boehm y Turner . La Habana, Cuba : Serie Científica de la Universidad de las Ciencias Informáticas., 2012, Vol. 5.

ALBERT, JUAN SILVIO CABRERA. El estudio de los estilos de aprendizaje: desde una perspectiva vigostkiana: aproximación conceptual. Pinar del Río, Cuba : Revista Iberoamericana de Educación. 1681-5653.

Algoritmos Conceptuales: Una perspectiva para el modelado del estudiante en los Sistemas Tutoriales Inteligentes. La Habana : Revista de Ciencias Informáticas.

González ,Yunia R., Arceo, Alfonso C., Sánchez, Natalia M., Domínguez, Antonio H., 2016. Agrupamiento conceptual lógico combinatorio: una alternativa para la toma de decisiones. La Habana, Cuba: IBERAMIA and the authors, Vol. 19. ISSN: 1988-3064(on-line).

D. Pascual 1, F. Pla 2, S. Sánchez .Algoritmos de agrupamiento. Santiago de Cuba, Cuba : s.n.

ANEXO 1

Entrevista

Nombre y Apellidos: _____

Categoría docente: _____ **Años de experiencia:** _____

Categoría científica: _____

Especialidad: _____

Centro al que pertenecen: _____

Facultad: _____

Esta encuesta tiene como objetivo determinar el estilo de aprendizaje, las competencias y el tipo de inteligencia que debe predominar en los estudiantes de ingeniería para la selección de las estrategias de enseñanza y el desarrollo de las estrategias educativas de brigada.

Se les agradece de antemano por su colaboración.

1. De los estilos de aprendizaje siguientes:

Señale cuáles conoce		De los conocidos, establezca un orden de prioridad según su importancia. 1 (Mínima importancia) 5 (Máxima importancia)	
<input type="checkbox"/> Visual/ Verbal		<input type="checkbox"/> Visual/ Verbal	
<input type="checkbox"/> Secuencial/Global	<input type="checkbox"/> Asimilador	<input type="checkbox"/> Secuencial/Global	<input type="checkbox"/> Asimilador
<input type="checkbox"/> Auditivo/Musical	<input type="checkbox"/> Inductivo/Deductivo	<input type="checkbox"/> Auditivo/Musical	<input type="checkbox"/> Inductivo/Deductivo
<input type="checkbox"/> Activo/Reflexivo	<input type="checkbox"/> Sensitivo/Intuitivo	<input type="checkbox"/> Activo/Reflexivo	<input type="checkbox"/> Sensitivo/Intuitivo
<input type="checkbox"/> Social	<input type="checkbox"/> Pragmático	<input type="checkbox"/> Social	<input type="checkbox"/> Pragmático
<input type="checkbox"/> Individual	<input type="checkbox"/> Teórico	<input type="checkbox"/> Individual	<input type="checkbox"/> Teórico
<input type="checkbox"/> Divergente	<input type="checkbox"/> Convergente	<input type="checkbox"/> Divergente	<input type="checkbox"/> Convergente
<input type="checkbox"/> Físico/Kinestésico	<input type="checkbox"/> Acomodador	<input type="checkbox"/> Físico/Kinestésico	<input type="checkbox"/> Acomodador

1.1 ¿Considera usted que faltó por mencionar algún un estilo de aprendizaje?

De ser así diga cuál es y establezca su orden de prioridad según su grado de importancia.

2. Señale las competencias que considere que deba poseer un estudiante universitario de carreras de ingeniería.

- Capacidad de análisis y síntesis.
- Capacidad de aprender.
- Habilidad de resolver problemas.
- Capacidad de aplicar el conocimiento.
- Capacidad de adaptarse a situaciones nuevas.
- Preocupación de la calidad.
- Destrezas para manejar la información.
- Capacidad de trabajar en equipo.
- Capacidad de abstracción.
- Capacidad de comunicación oral y escrita.
- Capacidad de investigación.
- Capacidad de trabajar aplicar los conocimientos en la práctica.
- Capacidad para organizar y planificar el tiempo.
- Conocimientos sobre el área de estudio y la profesión.
- Capacidad de comunicación en un segundo idioma.
- Capacidad de aprender y actualizarse permanentemente.
- Habilidades para buscar, procesar y analizar información procedente de fuentes diversas.
- Capacidad crítica y autocrítica.
- Capacidad creativa.
- Compromiso ético.
- Compromiso con la calidad.

2.1 ¿Considera usted que faltó por mencionar alguna competencia? De ser así diga cuál es.

3. De los tipos de inteligencia siguientes:

Señale cuáles conoce	De los conocidos, establezca un orden de prioridad según su importancia 1 (Mínima importancia) 5 (Máxima importancia)
<input type="checkbox"/> Inteligencia musical.	<input type="checkbox"/> Inteligencia musical.
<input type="checkbox"/> Inteligencia lingüística.	<input type="checkbox"/> Inteligencia lingüística.
<input type="checkbox"/> Inteligencia espacial.	<input type="checkbox"/> Inteligencia espacial.
<input type="checkbox"/> Inteligencia interpersonal.	<input type="checkbox"/> Inteligencia interpersonal.
<input type="checkbox"/> Inteligencia intrapersonal.	<input type="checkbox"/> Inteligencia intrapersonal.
<input type="checkbox"/> Inteligencia naturalista.	<input type="checkbox"/> Inteligencia naturalista.
<input type="checkbox"/> Inteligencia lógico matemática.	<input type="checkbox"/> Inteligencia lógico matemática.
<input type="checkbox"/> Inteligencia corporal-kinestésica.	<input type="checkbox"/> Inteligencia corporal-kinestésica.

3.1 ¿Considera usted que faltó por mencionar algún tipo de inteligencia? De ser así diga cuál es y establezca su orden de prioridad según su grado de importancia.

Datos de los expertos.

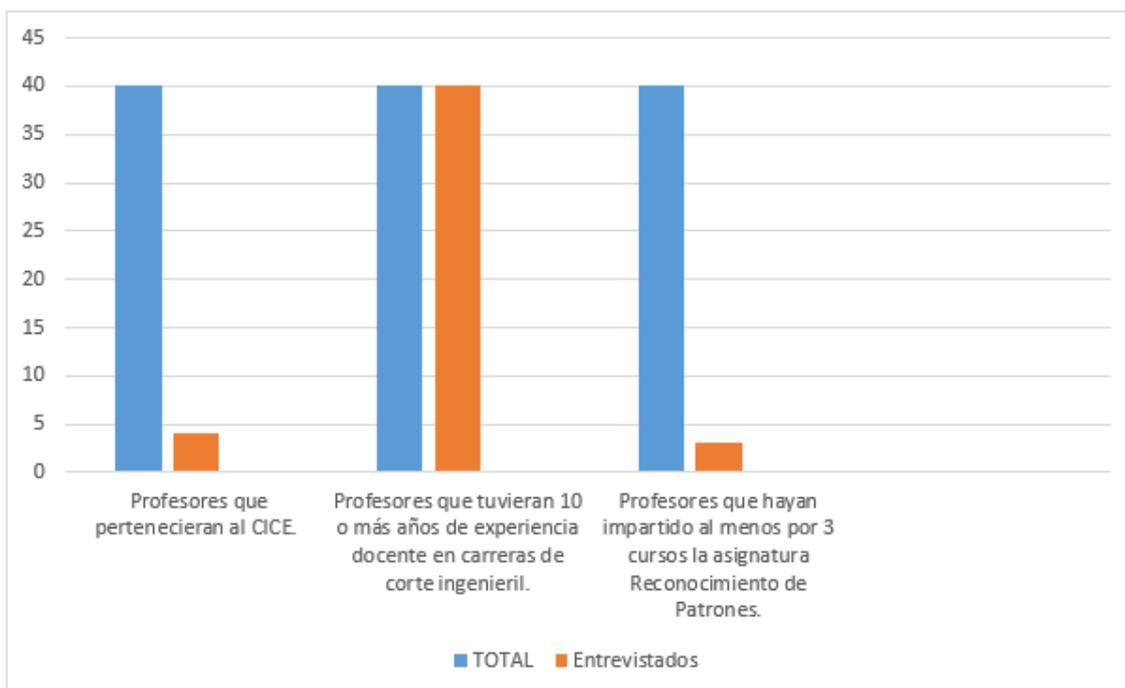


Figura 12 Datos de los expertos entrevistados

ANEXO 2

SECCIÓN 1

1. **Tengo que hallarle sentido a lo que hago, sino no me siento satisfecho.**
_0 _1 _2 _3 _4 _5
2. **Puedo sumar o multiplicar mentalmente con mucha rapidez**
_0 _1 _2 _3 _4 _5
3. **Encuentro fácil la resolución de problemas.**
_0 _1 _2 _3 _4 _5
4. **Con frecuencia veo configuraciones y relaciones entre números con más facilidad que otros.**
_0 _1 _2 _3 _4 _5
5. **La Matemática siempre ha sido una de mis asignaturas favoritas.**
_0 _1 _2 _3 _4 _5
6. **Me gustan los juegos y rompecabezas que requieran de la lógica.**
_0 _1 _2 _3 _4 _5

7. **Me gusta clasificar y jerarquizar cosas**
_0 _1 _2 _3 _4 _5
8. **No puedo comenzar una tarea sino tengo todos los pasos detallados para resolverla**
_0 _1 _2 _3 _4 _5
9. **Si algo se descompone y no funciona, tiendo a jugar con las piezas tratando de hacer que encajen.**
_0 _1 _2 _3 _4 _5
10. **Para una presentación grupal, organizo y clasifico la información en categorías**
_0 _1 _2 _3 _4 _5

SECCIÓN 2

1. **Puedo recordar mucho mejor los contenidos cuando los veo escritos.**
_no se ajusta en ninguna manera
_se ajusta solamente en algunos casos
_se ajusta completamente
2. **Me siento mucho mejor aprendiendo lo que puedo ver, a tener que aprender de lo que solamente escucho**
_no se ajusta en ninguna manera
_se ajusta solamente en algunos casos
_se ajusta completamente
3. **Me gusta que los profesores escriban en la pizarra, hagan esquemas, gráficos, muestren láminas, etc.**

- no se ajusta en ninguna manera
- se ajusta solamente en algunos casos
- se ajusta completamente

4. Para fijar un contenido nuevo, necesito representarlo y verlo en mi mente primero

- no se ajusta en ninguna manera
- se ajusta solamente en algunos casos
- se ajusta completamente

5. Me gusta aprender los contenidos que yo pueda observar, ya sea a través de imágenes, láminas, diagramas, gráficos, etc.

- no se ajusta en ninguna manera
- se ajusta solamente en algunos casos
- se ajusta completamente

6. No me gusta mucho escribir en clases, sólo seguir las explicaciones del profesor

- no se ajusta en ninguna manera
- se ajusta solamente en algunos casos
- se ajusta completamente

7. Me gusta estudiar con alguien que me lea los contenidos, yo los escucho y así me los grabo mejor.

- no se ajusta en ninguna manera
- se ajusta solamente en algunos casos
- se ajusta completamente

8. Me es fácil escuchar y seguir al profesor cuando explica en clases, lo comprendo todo, aun cuando no se escriba nada en la pizarra.

- no se ajusta en ninguna manera
- se ajusta solamente en algunos casos
- se ajusta completamente

9. Aprendo mucho mejor cuando me repito los contenidos a mí mismo(a), y si lo hago en alta voz, pues mucho mejor

- no se ajusta en ninguna manera
- se ajusta solamente en algunos casos
- se ajusta completamente

SECCIÓN 3

1. Entiendo mejor algo

- a) ___ si lo práctico.
- b) ___ si pienso en ello.

2. Cuando estoy aprendiendo algo nuevo, me ayuda

- a) ___ hablar de ello.
- b) ___ pensar en ello.

3. En un grupo de estudio que trabaja con un material difícil, es más probable que

- a) ___ participe y contribuya con ideas.
- b) ___ no participe y solo escuche.

4. En las clases a las que he asistido

- a) ___ he llegado a saber cómo son muchos de los estudiantes.
- b) ___ raramente he llegado a saber cómo son muchos estudiantes.

5. Cuando comienzo a resolver un problema de tarea, es más probable que

- a) ___ comience a trabajar en su solución inmediatamente.
- b) ___ primero trate de entender completamente el problema.

6. Prefiero estudiar

- a) ___ en un grupo de estudio.
- b) ___ solo.

7. Prefiero primero

- a) ___ hacer algo y ver qué sucede.
- b) ___ pensar cómo voy a hacer algo.

8. Recuerdo más fácilmente

- a) ___ algo que he hecho.
- b) ___ algo en lo que he pensado mucho.

9. Cuando escribo un trabajo, es más probable que

- a) ___ lo haga (piense o escriba) desde el principio y avance.
- b) ___ lo haga (piense o escriba) en diferentes partes y luego las ordene.

10. Me considero

- a) ___ abierto.
- b) ___ reservado.

SECCIÓN 4

Marca los enunciados que describen tu comportamiento

1. ___ Normalmente trato de resolver los problemas metódicamente y paso a paso
2. ___ Me interesa saber cuáles son los sistemas de valores de los demás y con qué criterios actúan.
3. ___ Estoy a gusto siguiendo un orden, en las comidas, en el estudio, haciendo ejercicio regularmente
4. ___ Normalmente encajo bien con personas reflexivas, y me cuesta sintonizar con personas demasiado espontáneas, imprevisibles.
5. ___ Prefiero las cosas estructuradas a las desordenadas.

6. ___ Casi siempre procuro ser coherente con mis criterios y sistemas de valores. Tengo principios y los sigo.
7. ___ Me disgusta implicarme afectivamente en mi ambiente de trabajo. Prefiero mantener relaciones distantes.
8. ___ Me cuesta ser creativo(a), romper estructuras
9. ___ Me molesta que la gente no se tome en serio las cosas.
10. ___ Esquivo los temas subjetivos, ambiguos y poco claros.
11. ___ Si trabajo en grupo procuro que se siga un método y un orden.
12. ___ Ante los acontecimientos trato de descubrir los principio y teorías en que se basan
13. ___ Me molestan las personas que no actúan con lógica.
14. ___ Con frecuencia miro hacia delante para prever el futuro.
15. ___ Observo que, con frecuencia, soy uno(a) de los(as) más objetivos(as) y desapasionados(as) en las discusiones
16. ___ Siempre trato de conseguir conclusiones e ideas claras
17. ___ Estoy convencido(a) que debe imponerse la lógica y el razonamiento.
18. ___ Detecto frecuentemente la inconsistencia y puntos débiles en las argumentaciones de los demás.
19. ___ Tiendo a ser perfeccionista.
20. ___ Estoy seguro(a) de lo que es bueno y lo que es malo, lo que está bien y lo que está mal.

ANEXO 3

Tareas de Ingeniería

Tabla 19 Tarea de ingeniería: Cálculo de la Matriz de Diferencia.

Tarea	
Número de la tarea: 10	Número de la historia: 9
Nombre de la tarea: Cálculo de la Matriz de Diferencia.	
Tipo de tarea: Implementación	Estimación: 5 días
Programador responsable: Beatriz Lugo - Liomar Rodríguez	
Descripción: Implementar los métodos necesarios para calcular la matriz de diferencia.	

Tabla 20 Tarea de ingeniería: Cálculo de la Matriz Básica

Tarea	
Número de la tarea: 11	Número de la historia: 9
Nombre de la tarea: Cálculo de la Matriz Básica	
Tipo de tarea: Implementación	Estimación: 5 días
Programador responsable: Beatriz Lugo - Liomar Rodríguez	
Descripción: Implementar los métodos necesarios para calcular la matriz básica.	

Tabla 21 Tarea de ingeniería: Selección de los testores.

Tarea	
Número de la tarea: 12	Número de la historia: 9
Nombre de la tarea: Selección de los testores.	
Tipo de tarea: Implementación	Estimación: 5 días
Programador responsable: Beatriz Lugo - Liomar Rodríguez	
Descripción: Implementar los métodos necesarios	

para seleccionar los testores.

Tabla 22 Tarea de ingeniería: Calcular relevancia de los rasgos.

Tarea	
Número de la tarea: 13	Número de la historia: 9
Nombre de la tarea: Calcular relevancia de los rasgos.	
Tipo de tarea: Implementación	Estimación: 5 días
Programador responsable: Beatriz Lugo - Liomar Rodríguez	
Descripción: Implementar los métodos necesarios para calcular la relevancia de los rasgos.	

Tabla 23 Tarea de ingeniería: Selección de los testores más relevantes.

Tarea	
Número de la tarea: 14	Número de la historia: 9
Nombre de la tarea: Selección de los testores más relevantes.	
Tipo de tarea: Implementación	Estimación: 5 días
Programador responsable: Beatriz Lugo - Liomar Rodríguez	
Descripción: Implementar los métodos necesarios para seleccionar los testores más relevantes.	

Tabla 24 Tarea de ingeniería: Generación de los I-complejos de cada grupo.

Tarea	
Número de la tarea: 15	Número de la historia: 9
Nombre de la tarea: Generación de los I-complejos de cada grupo.	
Tipo de tarea: Implementación	Estimación: 5 días
Programador responsable: Beatriz Lugo - Liomar Rodríguez	
Descripción: Implementar los métodos necesarios	

para generar los l-complejos de cada grupo.

ANEXO 4

Casos de Pruebas de Aceptación

Tabla 25 Caso de Prueba de Aceptación 13.

Caso de Prueba de Aceptación	
Código: P13	Número de la historia:9
Nombre: Agrupar estudiantes.	
Descripción: El administrador después de verificar que todos los estudiantes matriculados en el curso hayan respondido los cuestionarios y estén clasificados seleccionan la opción de "Agrupar estudiantes".	
Condiciones de ejecución: Debe estar corriendo el servidor Apache.	
Entradas/Pasos de ejecución: <ul style="list-style-type: none">1- Ejecutar la aplicación.2- Los estudiantes deben estar clasificados.	
Resultado esperado: Se espera que los estudiantes se agrupen en clases y se muestren los conceptos de cada clase y esos datos se almacene correctamente en la BD.	
Evaluación de la prueba: Prueba satisfactoria	