



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 1**

**Módulo estudiante para el Sistema Tutorial Inteligente en la enseñanza  
de la Inteligencia Artificial 1**

**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas**

**Autor:**

**Est. Noel Luis Martínez Ramírez**

**Tutores:**

**MSc. Maidelis Milanés Luque**

**Ing. Estela Odelsa Martín Coronel**

**La Habana, mayo de 2022**

**“Año 62 de la Revolución”**

## **DECLARACIÓN DE AUTORIA**

---

Declaro ser autor de la presente tesis que tiene por título Módulo estudiante para el Sistema Tutorial Inteligente en la enseñanza de la Inteligencia Artificial 1 y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Autor

\_\_\_\_\_  
Noel Luis Martínez Ramírez

Tutor

\_\_\_\_\_  
MSc. Maidelis Milanés Luque

Tutor

\_\_\_\_\_  
Ing. Estela Odelsa Martín Coronel

## **SINTESIS DEL TUTOR**

---

**Maidelis Milanés Luque:** Graduada de Ingeniera en Ciencias Informáticas en el 2007, actualmente realizando investigaciones en el área de la Inteligencia Artificial, específicamente en la Minería de Datos (Algoritmos de agrupamiento)

**Estela Odelsa Martín Coronel:** Graduada de Ingeniera en Ciencias Informáticas en el 2014, profesora de Inteligencia Artificial 2 e Introducción a las Ciencias Informáticas.

# ÍNDICE

---

INTRODUCCIÓN .....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA .....	5
1.1. Introducción .....	¡Error! Marcador no definido.
1.2. Sistemas Tutoriales Inteligentes: .....	5
1.3 Los componentes fundamentales de un STI:.....	5
1.4 Inteligencia artificial: .....	7
1.5 Sistemas Expertos:.....	8
1.6. Técnicas de inteligencia artificial: .....	9
1.7 Las Redes Neuronales Artificiales (RNA): .....	10
1.8 Software: .....	10
1.9 Proceso de enseñanza aprendizaje. Elementos:.....	11
1.10 Análisis a soluciones existentes: .....	13
1.11 Valoración de las herramientas:.....	15
1.12 Metodología de desarrollo de Software:.....	16
1.12.1 Metodología AUP-UCI: .....	17
1.12.2 Fases de AUP-UCI:.....	17
1.13 Lenguajes de Programación: .....	18
1.13.1 Lenguaje PHP:.....	19
1.14 Entorno de desarrollo integrado (IDE):.....	20
1.15 Marcos de Trabajo: .....	22
1.16 Gestor de Base de Datos:.....	23
1.17 Herramienta CASE:.....	24

# ÍNDICE

---

## CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL MÓDULO ESTUDIANTE PARA EL SISTEMA TUTORIAL INTELIGENTE. ....27

Introducción:.....¡Error! Marcador no definido.

2.1	Propuesta de solución: .....	27
2.2	Personas que interactúan con el sistema:.....	28
2.3	Modelo Conceptual:.....	29
2.4	Obtención de requisitos:.....	31
2.4.1	Técnicas de levantamientos de los requisitos de software: .....	31
2.5	Requisitos Funcionales: .....	32
2.6	Requisito no funcional: .....	34
2.7	Diagrama de Casos de Uso del Sistema:.....	35
2.8	Modelo de datos: .....	36
2.9	Arquitectura: .....	37
2.10	Patrones de Diseño: .....	38
2.11	Patrones para asignar responsabilidades:.....	39
2.13	Diagrama de clase del diseño:.....	41
2.14	Conclusiones Parciales: .....	42

## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO ESTUDIANTE PARA EL SISTEMA TUTORIAL INTELIGENTE. ....44

3.1.	Introducción:.....	¡Error! Marcador no definido.
3.2	Estándares de codificación: .....	44
3.3	Diagrama de componentes: .....	46
3.4	Diagrama de Despliegue: .....	47
3.5	Pruebas a realizar:.....	48

# ÍNDICE

---

3.6 Conclusiones parciales: .....	49
<b>CONCLUSIONES GENERALES:.....</b>	<b>50</b>
<b>RECOMENDACIONES:.....</b>	<b>51</b>
<b>REFERENCIAS BIBLIOGRÁFICAS:.....</b>	<b>52</b>
<b>Anexos: .....</b>	<b>55</b>

## AGRADECIMIENTOS Y DEDICATORIA

---

### Dedicatoria

A mi familia y amigos, por motivarme constantemente para alcanzar mis sueños y por el apoyo incondicional que me han brindado; muchos de mis logros se los debo a ustedes, en especial este.

A mi querido padre, quien a pesar de no haber podido estar a mi lado, siempre me acompañó en mi corazón y dio fuerzas para seguir adelante.

Este triunfo es de ustedes también.

## **AGRADECIMIENTOS Y DEDICATORIA**

---

### **Agradecimientos**

Mediante estos agradecimientos deseo exaltar la labor de todos aquellos que estuvieron presentes durante toda o mayor parte de la realización y desarrollo de esta tesis, a los que han sido claves en mi vida de preparación profesional y por extensión, en la personal. A todos muchas gracias.

A mis padres: Magda y Juan, por ser los principales promotores de mis sueños, por su crianza y valiosas enseñanzas que me han hecho la persona que soy y a su gran apoyo en el proceso de confección de esta tesis.

Agradezco también a mi hermano Juan Luis por su gran ayuda y guía en estos tiempos y toda mi vida.

Mil gracias a mis amigos: Reinier, Rocío, Alejandra, Alejandro, Riquene, Dariel y Álvaro por haberme ayudado de tantas maneras; a mis compañeros de la escuela por su maravillosa compañía durante estos últimos cinco años: Raidel, Lena, Yurena, Jessica, Mitchel y Julio, entre tantos otros; y a mi novia Ana por estar a mi lado y creer en mí.

A mis seres queridos, quienes representaron un gran apoyo emocional: Lisandra, Juan Carlos, Juana, Antonio, Maira, Dorin, mis suegros, Danays y Eduardo y a toda mi familia.

Quisiera agradecer en especial a cada uno de mis maestros a lo largo de estos años de estudio, personas de gran sabiduría que formaron parte de mi proceso integral de formación; no ha sido un camino sencillo, pero gracias a su afán de transmitirme su conocimiento y toda su dedicación, he logrado importantes objetivos, como fue culminar el desarrollo de mi tesis con éxito para finalmente obtener mi anhelado título profesional.

Gracias de corazón a mis tutoras: la MSc. Maidelis Milanés Luque e Ing. Estela Odelsa Martín Coronel, por su buena fe y sus respetuosos criterios y aportes a este trabajo. Ha sido un privilegio contar con su guía y ayuda.

A la Universidad de Ciencias Informáticas por todos los recursos que han puesto a mi disposición en todos estos años de estudio para permitirme llegar a donde he llegado, por todas mis experiencias, recuerdos y relaciones forjadas en esta magistral institución.



## **AGRADECIMIENTOS Y DEDICATORIA**

---

Finalmente agradezco a quien lee esta tesis, por permitir que mis experiencias, investigación y conocimiento trasciendan y resulten útiles para su propia preparación y la de otros.

### Resumen

El impacto de las Tecnologías de la Información y la Comunicación (TIC) toca muy de cerca a los sectores educacionales, propiciando modificaciones en las formas tradicionales de enseñar y aprender.

Con las transformaciones constantes en las TIC y la búsqueda de mejoras en los procesos docentes, han surgido nuevas estrategias en el proceso docente educativo, entre las que se encuentran los Sistemas Tutoriales Inteligentes (STI). Estos tienen como objetivo imitar a los tutores humanos en su habilidad para determinar en cada caso: qué enseñar, cuándo enseñar y cómo enseñar en lo posible de un modo autónomo.

La presente investigación plantea como objetivo desarrollar una herramienta informática que clasifique adecuadamente a los estudiantes que van a utilizar el STI para la enseñanza de la Inteligencia Artificial<sup>1</sup> en la Universidad de las Ciencias Informáticas, utilizando para ello las Redes Neuronales como técnica de Inteligencia Artificial (IA), PHP como lenguaje de programación del lado del cliente y del lado del servidor HTML, CSS y JavaScript, como framework de apoyo Symfony, además de IDE PhpStorm en su versión 7.1.3, PostgreSQL como gestor de base de datos y Apache como servidor web.

Con la futura implementación del sistema se pretende crear un vínculo más estrecho con el estudiante, conocer que estilo de aprendizaje se adecua más a él y de qué forma se puede motivar para que la enseñanza de la IA sea más próspera y sencilla, así fortalecer la trayectoria de los estudiantes durante su formación como futuros ingenieros informáticos

**Palabras clave:** Educación, Enseñanza, Inteligencia Artificial, Módulo estudiante, Sistemas Tutores Inteligentes.

## RESUMEN

---

### **Abstract**

The impact of Information and Communication Technologies (ICT) touches the educational sectors very closely, fostering changes in traditional ways of teaching and learning.

With the constant transformations in ICT and the search for improvements in teaching processes, new strategies have emerged in the educational teaching process, among which are Intelligent Tutoring Systems (ITS). These aim to imitate human tutors in their ability to determine in each case: what to teach, when to teach and how to teach in an autonomous way as much as possible.

The present research aims to develop a computer tool that adequately classifies the students who are going to use the ITS for the teaching of Artificial Intelligence 1 at the University of Computer Science, using Neural Networks as an artificial intelligence (AI) technique, PHP as a programming language on the client side and on the server side HTML, CSS and JavaScript, as support framework Symfony , as well as IDE PhpStorm in its version 7.1.3, PostgreSQL as database manager and Apache as web server.

With the future implementation of the system, it is intended to create a closer bond with the student, to know which learning style is best suited to him and how he can be motivated so that the teaching of AI is more prosperous and simple, thus we strengthen the students' trajectory during their training as future computer engineers

Keywords: Education, Teaching, Artificial Intelligence, Student module, Intelligent Tutor Systems, ICT

### INTRODUCCIÓN

El uso de las Tecnologías de la Información y las Comunicaciones (TIC), ha logrado un salto vertiginoso en el desarrollo científico técnico a escala mundial. Estas han mejorado la manera de trabajar y gestionar recursos en las organizaciones. En nuestro país se está llevando a cabo la importante tarea de informatizar la sociedad, promoviendo la utilización masiva de las TIC para satisfacer las necesidades de información y conocimiento de todas las personas, esferas de la sociedad y facilitar el Proceso de Enseñanza Aprendizaje (PEA).

En los últimos tiempos en la esfera de la educación se han desarrollado herramientas informáticas que favorecen el aprendizaje y el conocimiento humano, explotando todas las facilidades que brindan las tecnologías de punta al proceso de enseñanza-aprendizaje (Soledad González, C. 2004). La UCI cuenta con muchas herramientas tecnológicas destinadas al proceso docente-educativo, las cuales están diseñadas a apoyar y fortalecer la trayectoria de los estudiantes durante su formación como ingenieros informáticos. Entre las herramientas más importantes se encuentran los Sistemas Tutoriales Inteligentes (STI). Estos sistemas tienen como propósito exhibir un comportamiento parecido al de un tutor humano, por tal razón, tienen que ser capaces de adaptarse al comportamiento mostrado por un estudiante poseen actividades de retroalimentación y autoevaluación que informan sobre los avances en la ejecución y los logros de los objetivos educacionales que persiguen los educandos (Soledad González, C. 2004).

El modelo del alumno es el componente del STI que representa el estado actual del conocimiento del estudiante y el proceso que gestiona esta estructura se llama diagnóstico. Ambos componentes deben diseñarse juntos y este problema de diseño es el que se conoce como el problema del modelado del estudiante. Se considera que una de las cuestiones más complejas en el modelado del alumno es el diagnóstico inicial del estudiante, porque en el mismo se deben definir tanto las características esenciales que se deben tener en cuenta de este, como el conocimiento previo necesario para enfrentar la materia a vencer.

Actualmente, se desconocen cuáles son los rasgos de los estudiantes que se deben tener en cuenta para enseñar adecuadamente la asignatura Inteligencia Artificial 1, así como los conocimientos previos que deben dominar en otras áreas del aprendizaje.

A partir de lo expuesto anteriormente se define como problema a resolver: ¿Cómo clasificar adecuadamente a los estudiantes que van a utilizar el STI para la enseñanza de la Inteligencia Artificial 1 en la Universidad de las Ciencias Informáticas?

Para darle solución al problema anterior se propone como **objeto de estudio**: El modelado del estudiante en los Sistemas Tutoriales Inteligentes.

Para ello se plantea como **objetivo general**: Desarrollar una herramienta informática que clasifique adecuadamente a los estudiantes que van a utilizar el STI para la enseñanza de la Inteligencia Artificial 1 en la Universidad de las Ciencias Informáticas. Delimitando como **campo de acción**: El proceso de diagnóstico inicial dentro del modelado del estudiante en los STI.

**Para darle cumplimiento al objetivo planteado se responderán las siguientes preguntas científicas:**

- ¿Cuáles son los referentes teóricos que sustentan el desarrollo de los sistemas tutoriales inteligentes?
- ¿Qué aspectos deben tenerse en cuenta para realizar el análisis y diseño del módulo estudiante para el desarrollo de un STI como apoyo al proceso de enseñanza-aprendizaje de la Inteligencia Artificial I en la Universidad de las Ciencias Informáticas?
- ¿Cómo validar el adecuado análisis y diseño del módulo estudiantes después de identificados todos los requisitos del sistema?

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

- Estudio del modelado del alumno en los STI para asimilar las bases teóricas de la investigación.
- Análisis del componente de diagnóstico inicial en el modelado del alumno de diferentes STI con el propósito de reutilizar los elementos que puedan servir para el desarrollo de la herramienta.
- Definición de los rasgos cognitivos que se deben tener en cuenta en un estudiante para enseñar la asignatura Inteligencia Artificial 1.
- Definición de los conocimientos previos que debe poseer un estudiante que va a cursar la asignatura Inteligencia Artificial 1.

- Estudio de varias técnicas de IA en la clasificación para definir la posición del investigador.
- Selección de la metodología de desarrollo de software que sirva de guía para el proceso de elaboración de la herramienta.
- Selección de las herramientas, tecnologías, lenguajes, entre otros que permitan el desarrollo de la herramienta.
- Selección y realización de las pruebas que permitan verificar el correcto funcionamiento de la herramienta.
- Descripción de futuros desarrollos sobre la herramienta implementada.

Los **métodos de investigación** utilizados para dar solución a la presente investigación son:

### **Teóricos:**

**Histórico-Lógico:** Este método permite una mejor comprensión de la investigación mediante el estudio y profundización de los STI en el mundo, así como sus principales características e impacto en el PEA.

**Analítico-Sintético:** Este método permite determinar las principales técnicas de IA, herramientas y metodologías que son usados a nivel mundial en estos sistemas y que puedan ser de utilidad en la implementación del módulo.

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

### **Empíricos:**

**Entrevista:** Se emplea en encuentros con el cliente para conocer la necesidad del desarrollo de la propuesta de solución, definir sus funcionalidades e identificar a la vez particularidades de cada usuario y las restricciones que se imponen.

**Observación:** Posibilita obtener conocimiento acerca del funcionamiento de los sistemas existentes en la actualidad para proponer una mejor solución al proceso en cuestión.

**El contenido del presente trabajo de diploma está estructurado de la siguiente forma:**

**Capítulo 1- Fundamentación teórica:**

En este capítulo se realiza un estudio de los STI, específicamente el módulo estudiante definiendo su estructura y funcionamiento. Se explican diferentes conceptos esenciales para entender y darle solución al problema anteriormente planteado. Se realiza estudio de las principales metodologías, lenguajes, plataformas, herramientas, entre otros aspectos fundamentando en cada caso la elección con las que se llevará a cabo la propuesta de solución.

**Capítulo 2 – Análisis y Diseño del módulo estudiante para el sistema tutorial inteligente.**

En este capítulo se describe el funcionamiento del sistema propuesto guiado por la metodología AUP/UCI obteniéndose los artefactos de cada etapa. Se definen los casos de uso del sistema, los requisitos funcionales y no funcionales, el patrón arquitectónico y los patrones de diseño, los diagramas de diseño, el diagrama de despliegue y el diagrama de componentes.

**Capítulo 3 – Implementación y prueba del módulo estudiante para el sistema tutorial inteligente:**

Se abordan las etapas de diseño, implementación y pruebas donde se incluye dentro de la última mencionada la validación y verificación del sistema con las cuales se finaliza el proceso de desarrollo del software y por tanto la propuesta del sistema que presenta este trabajo.

## **CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA**

En este capítulo se ofrece un marco conceptual sobre el problema a resolver con el objetivo de profundizar en la investigación. Se abordan aspectos sobre los sistemas tutoriales inteligentes, se hace mención a algunos sistemas informáticos existentes a nivel nacional y mundial, dedicados a la realización de sistemas tutoriales inteligentes. Se analiza la metodología de desarrollo de software utilizada para la elaboración del sistema, analizando las etapas de diseño, implementación y pruebas. Se exponen y explican los lenguajes de programación, herramientas, y tecnologías a utilizar, en el desarrollo del sistema.

### **1.1. Sistemas Tutoriales Inteligentes:**

Un STI es capaz de guiar al alumno a lo largo de un dominio en particular del conocimiento, resolviendo durante el proceso tareas tales como la elaboración de una estrategia de tutorización, la generación de ejercicios a la medida de las necesidades del alumno, la resolución pedagógica de estos ejercicios, así como la explicación de la solución. Estas tareas se organizan en distintos módulos siendo los componentes claves del STI tradicional: un modelo del alumno, un modelo pedagógico, un modelo didáctico y una interface con la que interactúa el usuario. Dependiendo de la arquitectura del sistema estos módulos se pueden encontrar organizados en diferentes formas. Pueden estar distribuidos y subdivididos en partes más pequeñas, funcionando como entidades semi o completamente autónomas, que se comunican entre sí y actúan racionalmente de acuerdo a sus percepciones del exterior y el estado de su conocimiento. Podríamos decir que esta es una arquitectura basada en agentes inteligentes. Además, existen otros tipos de tutores inteligentes los cuales se basan en estos módulos, pero crean otros módulos personalizados. (Soledad González, C. 2004).

### **1.3 Los componentes fundamentales de un STI:**

El siguiente esquema representa los cuatro módulos principales que componen un STI:



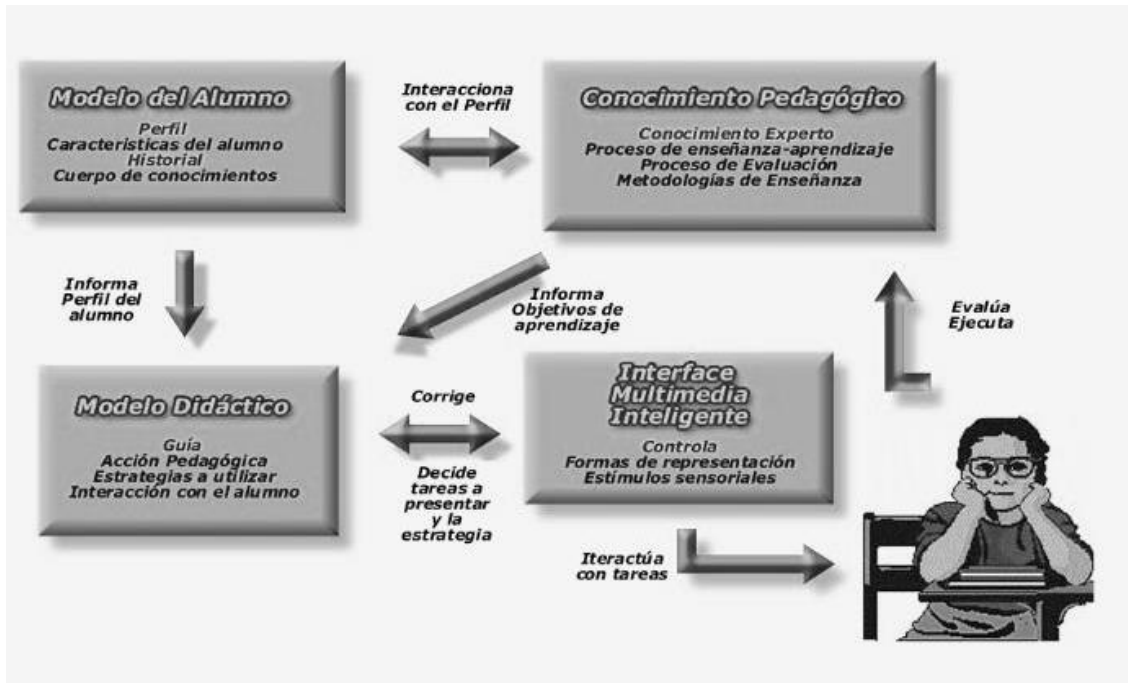


Figura 1: Modelo Tradicional De un STI (Cabimas, mayo 2012).

**El módulo "Modelo del alumno":** contiene el cuerpo de conocimientos que caracterizarán al usuario y lo representa desde perspectivas diferentes como, los aspectos psico-sociológicos característicos que condicionan el proceso de aprendizaje, el conocimiento que éste tiene sobre el dominio del tema a tratar y las destrezas y habilidades mínimas que debe tener para realizar una tarea. Este modelo además debe ser capaz de recoger el comportamiento evolutivo del alumno durante el trabajo en diferentes sesiones y modelar el estado mental del alumno, es decir "lo que sabe y lo que no sabe" y a partir de esto adaptar el sistema sobre la base de sus respuestas. Al ser este módulo el que representa el estado cognitivo del alumno, en la siguiente sección nos ocuparemos del problema del modelado en particular.

**El módulo "Pedagógico":** contiene representación de conocimiento experto en los ámbitos relativos a procesos de evaluación, enseñanza-aprendizaje; aprendizaje humano y metodología de enseñanza. Será el razonador, dónde se almacenará la base de conocimiento y los mecanismos de resolución de

problemas. Este módulo es el responsable de dirigir la ejecución del módulo "Modelo Didáctico" teniendo en cuenta los datos ingresados desde el módulo "Modelo del alumno".

**El módulo "Modelo Didáctico":** cumple la función de tutor o profesor y contiene información para decidir qué tareas se le presentan al estudiante de acuerdo con los objetivos de aprendizajes que el "Módulo Pedagógico" deja establecidos y los mecanismos para corregir el modelo del alumno. Es el encargado de generar los planes instruccionales de cada sesión. Este módulo es responsable de la activación del módulo "Interface".

Las tareas de aprendizaje son presentadas por el STI a través de una Interface Multimedia. Ésta debe estar dotada de múltiples medios de comunicación eficazmente integrados y combinados, para lograr una enseñanza adaptada y eficiente.

**El módulo "Interface-Multimedia"** contiene los mecanismos de representación (imágenes animadas, imágenes estáticas, sonido, lenguaje oral, lenguaje escrito, reconocimiento de voz, etc.) de informaciones necesarias para la realización de tareas que el sistema propone al sujeto. El éxito de un programa educativo, su calidad y efectividad, dependen en gran parte de la riqueza comunicadora que reúna. Esta cuestión empezó a interesar al campo de la Psicología Cognitiva, existiendo recientes investigaciones que demuestran que es fundamental tener en cuenta algunos principios de diseño multimedia para lograr y potenciar los aprendizajes (Moreno y Mayer, 2000).

### 1.4 Inteligencia Artificial (IA):

La IA es un campo de estudio que busca explicar y emular inteligencia, desarrollándola en términos de procesos computacionales que si son utilizados correctamente por un programa, puede exhibir un comportamiento inteligente. La IA pretende modelar a través de computadores la inteligencia del ser humano para resolver problemas de manera inteligente. Esta pretensión se basa en la experiencia que el hombre ha acumulado en el campo de su acción específica a través del aprendizaje. Las personas infieren y aprenden, y esto lo convierten en conocimiento que se registra en redes neuronales a través de interacciones sinápticas. La IA imita estos aprendizajes del hombre creando modelos para la solución de problemas en aplicación de elementos de lógica booleana, lógica difusa, cálculo infinitesimal, teorías de árboles, algoritmos de búsqueda, técnicas de optimización, teorías de aprendizaje, conceptos sobre grafos, arquitecturas computacionales y paradigmas para desarrollo de software.

El desarrollo de la IA ha tenido contrastes e históricamente ha tenido épocas de estancamiento. Sobre esto han influido aspectos como:

- Su denominación misma: ¿inteligencia artificial? ¿hombre artificial?
- El conocimiento incompleto sobre los procesos mentales del hombre.
- Sabemos que pensamos, pero no podemos explicar cómo lo hacemos de manera exacta, ni por qué.
- La imposibilidad matemática de expresar en ecuaciones, derivadas, integrales o algoritmos, modelos que expliquen comportamientos humanos.
- La incapacidad de los sistemas de cómputo para almacenar e inferir sobre la amplia gama de posibilidades, probabilidades, combinaciones y estados que pueden surgir del estudio de un problema cuya solución requiere del agregado inteligente.

Esta última dificultad se ha venido superando en las últimas décadas con los aportes de la electrónica a la capacidad de los dispositivos de almacenamiento, la ampliación de la memoria del computador para procesos más rápidos y con las redes de banda ancha. Con esto países como Japón han retomado las investigaciones sobre IA.

El propósito de la IA no es la creación de agentes u hombres artificiales, como en la ciencia ficción. Tampoco intenta crear hombres o artefactos con sus mismas sensibilizaciones. Las propuestas sobre su desarrollo apuntan a la creación de modelos para soluciones inteligentes de problemas en dominios específicos. El propósito en realidad es la creación de sistemas inteligentes, empleando este concepto, no en toda su extensión, sino en lo referente a lo cognitivo. (Castrillón, 2004).

### **1.5 Sistemas Expertos:**

En la actualidad existen aplicaciones para la gestión de los servicios de guardia. Con el objetivo de identificar ventajas y desventajas de estas, se realiza el siguiente análisis haciendo énfasis en las características y funcionalidades de este tipo de herramientas.

Los sistemas expertos son un área de la IA y sus objetivos tienen que ver con la imitación de un experto humano en la realización de una tarea específica. Un sistema experto es un programa de

computador que resuelve problemas de manera similar a como lo hace un ser humano. Para esto utiliza el siguiente modelo:

- Almacenamiento de experiencias en una base de conocimientos.
- Realización de inferencias sobre esos conocimientos, cuando una situación nueva le llega, o sea, cuando se le presenta algún problema.

Así proceden los expertos humanos, almacenan información o experiencias y luego las aplican cuando se les presenta un problema. En universidades de la ciudad de Medellín se han desarrollado proyectos de sistemas expertos para la solución de problemas específicos. Uno de estos fue un sistema experto que almacenaba a manera de reglas y proposiciones las normas, experiencias y conocimientos de un abogado, para luego diagnosticar y dar soluciones en materia de derecho laboral.

Un sistema experto tiene dos grandes módulos, la base de conocimientos en la cual se almacenan las experiencias y el motor de inferencia que son los programas que analizan, hacen diagnósticos y tomas decisiones. Para los propósitos educativos su importancia reside en que un docente es un experto en enseñanza y esto lleva a que en la construcción de sistemas tutoriales inteligentes sea necesario integrar programas cuyas arquitecturas sean sistemas expertos. Así un STI puede obedecer a la interacción de varios sistemas expertos. La dificultad principal reside en el problema de representar el conocimiento del experto. Además del dominio de conocimiento específico, las estrategias para la resolución de los problemas del experto también deben capturarse y codificarse de manera conveniente. Es posible que el experto use estas habilidades de manera inconsciente, por lo que el proceso de formalizar el conocimiento (hacer patente un dominio) es un objetivo que constituye un reto y que requiere la intervención del ingeniero del conocimiento para poder codificar el conocimiento del experto. (Ing. Eucario Parra Castrillón, 2004).

### **1.6. Técnicas de inteligencia artificial:**

Uno de los más rápidos y sólidos resultados que surgieron en las investigaciones de la IA fue que la inteligencia necesita conocimiento. El conocimiento posee algunas propiedades poco deseables pues es voluminoso, es difícil caracterizarlo con exactitud, cambia constantemente, y se distingue de los datos en que se organiza de tal forma que se corresponde con la forma en que va a ser usado. Con

los puntos anteriores se concluye que una técnica de IA es un método que utiliza conocimiento y debe ser comprendido por las personas que lo proporcionan.

En muchos programas, los datos pueden adquirirse automáticamente en muchos dominios de la IA, la mayor parte del conocimiento que se suministra a los programas lo proporcionan personas, haciéndolo siempre en términos que ellos comprenden, puede modificarse fácilmente para corregir errores, y puede usarse en gran cantidad de situaciones aún cuando no sea totalmente preciso o completo.

### 1.7 Las Redes Neuronales Artificiales (RNA):

Las Redes Neuronales Artificiales, ANN (Artificial Neural Networks por sus siglas en inglés) están inspiradas en las redes neuronales biológicas del cerebro humano. Están constituidas por elementos que se comportan de forma similar a la neurona biológica en sus funciones más comunes. Estos elementos están organizados de una forma parecida a la que presenta el cerebro humano. Las ANN al margen de "parecerse" al cerebro presentan una serie de características propias del cerebro. Por ejemplo, las ANN aprenden de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos. Aprender: adquirir el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Las ANN pueden cambiar su comportamiento en función del entorno. Se les muestra un conjunto de entradas y ellas mismas se ajustan para producir unas salidas consistentes. Generalizar: extender o ampliar una cosa. Las ANN generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, dentro de un margen, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de ruido o distorsión. Abstractar: aislar mentalmente o considerar por separado las cualidades de un objeto. Algunas ANN son capaces de abstraer la esencia de un conjunto de entradas que aparentemente no presentan aspectos comunes o relativos (REDES-NEURONALES-ARTIFICIALES-Y-SUS-APLICACIONES, Xabier Basogain Olabe, 2008).

### 1.8 Software:

Software es un término que hace referencia a un **programa informático**. Estas herramientas tecnológicas disponen de distintas aplicaciones que posibilitan la ejecución de una variada gama de tareas en un ordenador (computadora). (Julián Pérez Porto y Ana Gardey, 2014).

### 1.8.1 Software educativo:

Un software educativo es un programa informático que se emplea para educar al usuario. Esto quiere decir que el software educativo es una herramienta pedagógica o de enseñanza que, por sus características, ayuda a la adquisición de conocimientos y al desarrollo de habilidades.

Existen diferentes clases de software educativo. Algunos de estos programas son diseñados como apoyo al docente. De esta manera, el maestro o el profesor acude al software para ofrecer sus lecciones o para reforzar una clase. Otros tipos de software educativo, en cambio, se orientan directamente al alumno, ofreciéndole un entorno en el cual puede aprender por su propia cuenta.

El software educativo es muy importante en la educación a distancia. Estas herramientas tecnológicas permiten simular las condiciones que existen en un aula o un salón de clase. Así el estudiante puede “*ingresar*” a un salón virtual, interactuar con el docente a través de videoconferencias, chat o correo electrónico, completar evaluaciones, etc.

Dentro del entorno de un aula virtual pueden existir diferentes herramientas que faciliten la enseñanza y la comunicación entre todos los participantes. A pesar de contener el término «aula», es importante comprender que este tipo de espacio debe presentar ciertas diferencias con un salón convencional, para hacer frente a potenciales inconvenientes de tipo técnico, como ser la inestabilidad de las conexiones a Internet de los estudiantes, o bien la falta de potencia de sus equipos.

Dado que el software educativo persigue la integración como uno de sus objetivos primordiales, no se basa en tecnología de punta que tan sólo unos pocos puedan disfrutar, sino que utiliza recursos al alcance de la mayoría para brindar una experiencia rica y eficiente para todos.

(Julián Pérez Porto y Ana Gardey, 2014).

### 1.9 Proceso de enseñanza aprendizaje. Elementos.

Se entiende por PEA como el sistema de comunicación intencional que se produce en un marco institucional en el que se generan estrategias encaminadas a provocar el aprendizaje.

El aula es el lugar en el cual se reúnen los cuatro elementos claves del PEA: el alumno, el docente, la información que se va a comunicar y el medio. En la figura 2 se muestra esta relación.



**Figura 2. Relación entre los elementos del aprendizaje** (Benítez ,2007).

El alumno, es poseedor de estrategias de aprendizaje adquiridas durante su vida académica, es poseedor de un perfil social y emocional que lo diferencia de sus compañeros, además es el elemento central del PEA. El docente, es poseedor del conocimiento y de las estrategias pedagógicas, es el motivador, administrador de la información y de los elementos que integran el medio que rodea al alumno. La información está delimitada por los planes y programas de estudio.

El medio, está constituido por la infraestructura como: laboratorios, talleres, herramientas informáticas, medios de comunicación, recursos del aula, y otras herramientas que están a disposición de los alumnos para facilitar el aprendizaje y para mejorar la actividad social entre ellos. El medio en el cual se ve envuelto el aprendizaje, ha ido evolucionando a lo largo del tiempo, lo que ha redefinido el rol del docente, este nuevo rol le permite ser el administrador del conocimiento, el facilitador del aprendizaje, y el motivador que trabaja constantemente sobre las expectativas de los alumnos.

De acuerdo con lo expuesto, se considera que el acto de enseñar y aprender no debe denominarse como principios para la enseñanza, sino que debe incluir al alumno y permitirle al educador dirigir científicamente el desarrollo integral de la personalidad de estos, considerando sus estilos de aprendizaje en medios propicios para la comunicación y la socialización, en lo que el marco del aula de clases se extienda al entorno, la familia y la comunidad en general (Benítez ,2007).

### **1.10 Análisis a soluciones existentes:**

Algunas tecnologías como los agentes inteligentes, las redes neuronales, los sistemas expertos, los sistemas tutores inteligentes, que hace unos años solo existían en forma teórica o en los ámbitos universitarios, se utilizan a diario en aplicaciones de uso relativamente sencillo que se encuentran en la etapa de producción y no son solo casos de estudio en los laboratorios universitarios. Además, estas tecnologías están disponibles para las grandes empresas y los centros de alta tecnologías, así como también para aplicaciones pequeñas y medianas que pueden utilizarlas a diario (Salgueiro, 2010).

#### **1.10.1 Soluciones existentes internacionales:**

Con el avance de las tecnologías se encuentran muchísimos STI en todo el mundo, unos al alcance de todos, otros con acceso más restringido, pero todos ofreciendo servicios en los diferentes destinos. A continuación, enunciamos algunos ejemplos de soluciones existentes de STI.

##### **Herramienta de Autoría (HEDEA):**

Es una herramienta de autoría que trabaja con un laboratorio virtual y permite desarrollar un STI a partir de la definición temática de un curso. Dichos STI se basan en modelos probabilísticos partiendo del temario pesado de un curso. La generación del modelo del estudiante se realiza de manera automática y es transparente al usuario. Los STI generados por la herramienta son asociados a un laboratorio virtual existente. El modelo del estudiante toma en cuenta los valores de experimentos previos lo cual permite darle mayor valor al historial o a su último resultado (Cataldi, 2010).

##### **CTAT (Cognitive Tutor Authoring Tools)**

En la Universidad de Carnegie Mellon se desarrolló una herramienta para la construcción automática de tutores inteligentes llamada CTAT [Koedinger y col., 2004]. CTAT Es un grupo de herramientas que permite aprender bajo el paradigma de aprendizaje mediante la práctica o aprendizaje activo. CTAT permite la creación de tutores flexibles para tareas simples y complejas, es capaz de soportar múltiples estrategias que los estudiantes pueden utilizar en la solución de problemas. Adicionalmente, CTAT permite registrar la actividad y los pasos seguidos por los estudiantes y proveer ayuda y retroalimentación en tiempo real. CTAT está basado en la arquitectura cognitiva ACT-R [Anderson y col. 1997] que permite simular y entender el proceso cognitivo humano. ACT-R intentan entender de qué manera se organiza y se produce el conocimiento en los seres humanos. Conforme se desarrolla



la investigación en este tema, ACT-R ha evolucionado hacia el desarrollo de sistemas capaces de realizar un amplio rango de tareas cognitivas humanas, capturando en gran detalle la forma como el ser humano percibe, piensa y actúa en la solución de problemas (Borst y Anderson, 2015).

### **SircSim**

Fue desarrollado en conjunto por el departamento de ciencias de la computación del instituto de Illinois Instituto de Tecnología y el departamento de Fisiología de Rush Colegio de Medicina. Este tutor es el más avanzado actualmente en su tipo y se utiliza para complementar las clases teóricas sobre problemas cardiovasculares está constituido por los cuatro módulos de un STI, los módulos dominio, dominio, estudiante y la interfaz, utiliza redes neuronales como técnicas de IA (Cataldi, 2010).

#### **1.10.2 Soluciones existentes nacionales:**

##### **El Sistema Tutorial Inteligente para Diagnóstico y Tratamiento de Infecciones de Transmisión Sexual (STIITS):**

Fue desarrollado en la universidad de Cienfuegos en conjunto con especialistas de segundo grado del hospital Dr. Gustavo Aldereguia Lima. Que hicieron labor de expertos en la validación de la base de conocimientos para el posterior diagnóstico de las enfermedades, así como aportar todos los datos para la realización del tutorial. STIITS responde a una necesidad social de la provincia de Cienfuegos, y también nacional por sus posibilidades de empleo en la docencia. No existe algún otro software en el país con características similares para el mismo fin, está constituido por los cuatro módulos de un STI, los módulos dominio, tutor, estudiante y la interfaz y utiliza razonamiento basado en casos como técnica de IA (Maribel García, 2012).

##### **Sistema Experto para el diagnóstico médico de las enfermedades genéticas con Dismorfias (SEGEDIS):**

Creado con el objetivo de proporcionar a los genetistas cubanos una herramienta en el apoyo a sus decisiones. Además, permite a los genetistas del Centro Nacional de Genética Médica (CNGM) que poseen los permisos de administración sobre la misma, agregar nuevas enfermedades, actualizando sistemáticamente la aplicación, y de esta forma brindar una mejor atención a los pacientes. (Maribel García, 2012).

**SEDIM-SV:**

Utilizando la metodología Weiss-Kulikowski. Este es un Sistema Experto de Diagnóstico Médico de Sepsis Vaginales (SEDIM-SV), como medio de ayuda, para la consulta de Ginecología de la Universidad de las Ciencias Informáticas (García, 2012).

**VIRTEVALL:**

Metodología de evaluación para aprendizaje autónomo de idiomas extranjeros: este STI prevé crear un ambiente alternativo de evaluación para aprendizaje autónomo de idiomas extranjeros, que favorezca el trabajo de los estudiantes de manera independiente en sus debilidades, y profundizar en temas que sean de su interés en el aprendizaje del idioma inglés. Fue creado en el 2009 en la Universidad de las Ciencias Informáticas. (Idorka Morales, 2012).

**HESEI:**

Es una herramienta computacional para elaborar sistemas de enseñanza aprendizaje inteligente, la cual utiliza técnicas de IA y mapas conceptuales, con el objetivo de adaptar el sistema de enseñanza-aprendizaje, a través de una interfaz visual, a las características del alumno. Permite a expertos no especialistas en computación crear sistemas de enseñanza aprendizaje inteligentes que aborden contenidos de cualquier especialidad y nivel de enseñanza. Esta herramienta fue creada en la Universidad de Villa Clara (Idorka Morales, 2012).

**STI-SO:**

Desarrollado en la Universidad de las Ciencias Informáticas, es una herramienta de apoyo al proceso de enseñanza-aprendizaje de la asignatura Sistema Operativo, la cual ayuda a consolidar los contenidos del plan de estudio definido para la asignatura. El STI-SO fue estructurado en cuatro módulos: Módulo del Estudiante, Módulo del Dominio, Módulo Pedagógico y la Interfaz. Este STI utiliza el RBC como técnica de IA para asignarle al estudiante el material de estudio (Luisa Idorka Morales, 2012).

**1.11 Valoración de las herramientas:**

Con la investigación realizada a cada uno de los sistemas antes expuestos se pudo constatar que todos ayudan en diferentes áreas de la educación, salud, telecomunicaciones, bioinformáticas y otras vinculadas con diferentes instituciones, demostrando la gran importancia de los STI en el mundo actual

y dentro de este específicamente en el proceso enseñanza-aprendizaje. Por lo general todos estos cuentan con los cuatros módulos fundamentales que componen un STI, módulo tutor , módulo dominio , módulo estudiantes y módulo interfaz ,a pesar de que ninguno de ellos contribuye a la mejoría de PEA en la asignatura Inteligencia Artificial 1, su análisis fue fundamental en la búsqueda e identificación de las características y elementos necesarios a tener en cuenta para la solución de nuestro objetivo general ya que estudiamos cada una de las técnicas utilizadas para al final escoger, Redes Neuronales, el equipo tenía algo de experiencia de trabajos anteriores y se acopla perfectamente para la realización de nuestro STI.

### 1.12 Metodología de desarrollo de Software:

Una metodología de desarrollo de software no es más que un conjunto de procedimientos, técnicas, herramientas y un soporte documental que guía a los desarrolladores en la realización del nuevo software. (Pressman, 2002) Debido a la responsabilidad que tiene sobre la vida del software es necesario que sea elegido el enfoque adecuado para el desarrollo eficiente de este. Existen dos enfoques: los ágiles y los tradicionales o prescriptivos.

Para la selección del enfoque se realiza el método Boehm y Turner (Boeras Velázquez, y otros, 2012) también conocido como método de la estrella. Este caracteriza al proyecto de software a partir de 5 criterios y realiza una estimación acerca de cuan ágil o prescriptivo debe ser el enfoque a utilizar. Tiene 5 ejes, en cada uno se coloca un criterio, estos se explican a continuación (Boeras Velázquez, y otros, 2012): Tamaño: Este criterio se utiliza para representar el número de personas involucradas en el proyecto. Pueden tenerse en cuenta el nivel de complejidad que pueda presentarse en la comunicación entre los miembros del proyecto y los costos que pueden provocar cambios esperados.

**Criticidad:** Se utiliza para evaluar la naturaleza del daño ocasionado por defectos que no hayan sido detectados al producto. Su evaluación puede ser cualitativa.

**Dinamismo:** Representa la rapidez con la que pueden estar cambiando los requerimientos del proyecto.

**Personal:** Representa la proporción del personal con experiencia alta, media y baja. Los métodos orientados al plan no se ven afectados negativamente por este factor pues no interesa el nivel de experiencia con la que cuenten los miembros del equipo.

**Cultura:** Las organizaciones y las personas que relaciona el proyecto pueden depender de la confianza o de la relación contractual. Esto refleja el nivel de ceremonia necesario y aceptado:

documentación, control, formalismo en las comunicaciones. Es importante tener en cuenta el comportamiento de cada criterio en el proyecto, en correspondencia de los valores que alcancen se determinará el enfoque, mientras más cerca esté del centro el enfoque será el ágil, de lo contrario debe usarse un enfoque tradicional.

### **1.12.1 Metodología AUP-UCI:**

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, para ello nos apoyaremos en el Modelo CMMI-DEV v1.3 el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad.

Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además las buenas prácticas que define CMMI-DEV v1.3 se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajo se redujo a 1 la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a 11 (Tamara, R. S. 2015).

### **Fases de AUP-UCI:**

La Metodología Variación de AUP para la UCI está formada por tres fases, (Inicio, Ejecución y Cierre) para el ciclo de vida de los proyectos de la universidad, las cuales contienen las características de las cuatro fases (Inicio, Elaboración, Construcción y Transición) propuestas en AUP. Las características de las fases de la metodología de la universidad son:

**Inicio:** Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

**Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se

implementa y se libera el producto. Durante esta fase el software es transferido al ambiente de los usuarios finales o entregado al cliente junto con la documentación técnica. Además, en esta transición se capacita a los usuarios finales sobre la utilización de la aplicación.

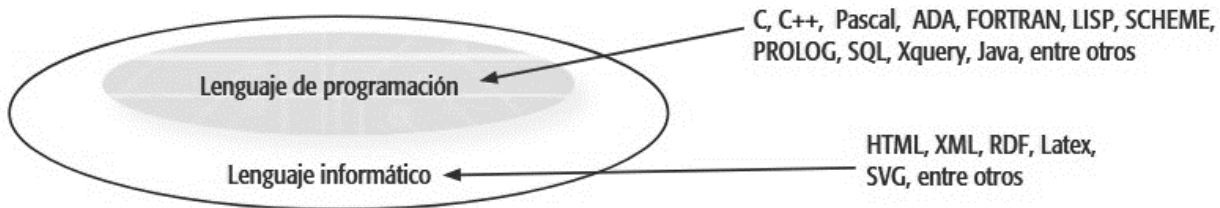
**Cierre:** El cierre se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto. (Tamara, R. S. (6 de marzo de 2015))

### **1.12.3 Fundamentos de la selección:**

AUP-UCI fue seleccionada debido a que se centra en equipos de desarrollo pequeño, como el presente caso que está integrado solamente por un programador de mediana experiencia. Además, brinda una gestión de cambios ágil que hace que el equipo de desarrollo se adapte a nuevas condiciones, que en este caso los requisitos pueden cambiar a decisión del cliente. Otro punto a su favor es la forma en que se planifica el proyecto y realiza la estimación del tiempo, pues se cuenta con poco tiempo para el desarrollo de la aplicación. Describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos. Es un marco de desarrollo de software iterativo e incremental. Finalmente, como factor más determinante de esta selección es que esta es una variación de la metodología AUP, que se adapta al ciclo de vida definido para la actividad productiva de la UCI.

### **1.13 Lenguajes de Programación:**

Un programa se escribe con instrucciones en un lenguaje de programación el cual a su vez está definido por su sintaxis, que establece e indica las reglas de escritura (la gramática), y por la semántica de los tipos de datos, instrucciones, definiciones, y todos los otros elementos que constituyen un programa. Un lenguaje de programación es un caso particular del lenguaje informático; este último permite hacer programas, pero también describir datos, configuraciones físicas y protocolos de comunicación entre equipos y programa.



**Figura.3 Tipos de lenguajes (Mihaela Juganaru Mathieu,2014).**

Si un programa está escrito en un lenguaje de programación comprensible para el ser humano, se le llama código fuente. A su vez, el código fuente se puede convertir en un archivo ejecutable (código máquina) con la ayuda de un compilador, aunque también puede ser ejecutado de inmediato a través de un intérprete (Mihaela Juganaru Mathieu,2014).

### 1.13.1 Lenguaje PHP:

El lenguaje de programación seleccionado fue PHP en su versión 5.4.12. Este es un lenguaje interpretado de alto nivel impregnado en páginas HTML y ejecutado en el servidor. PHP inició como una modificación a Perl a finales de 1994. Está orientado al desarrollo de aplicaciones web y permite insertar contenidos dinámicos en las páginas. PHP 5 se apoya en el llamado Zend Engine 2, la nueva versión del motor Zend que es el corazón de PHP desde la versión 4. El principal objetivo de PHP5 ha sido mejorar los mecanismos de la Programación Orientada a Objetos (POO), para solucionar las carencias de las anteriores versiones. Un paso necesario para conseguir que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos, incluso los más exigentes. (Rodríguez. O, 2010).

#### Características Generales del lenguaje.

- **Multiplataforma:** PHP funciona tanto en sistemas Unix o Linux con servidor web Apache como en sistemas Windows, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra.
- **Ejecución en Servidor:** Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a Bases de Datos, conexiones en red, y otras tareas

para crear la página final que verá el cliente.

- **Licencia de software libre:** PHP es un lenguaje basado en herramientas con licencia de software libre lo que significa que no hay que pagar licencias ni se está limitado su distribución y es posible ampliarlo con nuevas funcionalidades si así se desea.
- **Sintaxis cómoda:** PHP cuenta con una sintaxis similar a la de C, C++ o Perl. Lo más destacado ocurre a nivel semántico: el tipado es muy poco estricto. Se hace referencia específicamente a que, cuando creamos una variable no tenemos que indicar de qué tipo es, pudiendo guardar en ella datos de cualquier tipo.
- **Soporta objetos y herencia:** PHP tiene soporte para la POO, lo que permite crear clases para la construcción de objetos, con sus constructores, etc. Además, soporta herencia, aunque no múltiple.
- **Extensa biblioteca de funciones:** PHP cuenta con una extensa biblioteca de funciones que facilitan enormemente el trabajo de los desarrolladores.
- **Compatibilidad con Bases de Datos:** Una de las características más significativas de PHP es su amplio soporte para una gran cantidad de Bases de Datos. Tiene acceso a un gran número de gestores de Bases de Datos entre las que se encuentran MySQL, PostgreSQL, etc.
- **Expansión:** PHP está alcanzando unos niveles de uso tan elevados que hacen que su conocimiento sea algo indispensable para los profesionales del desarrollo en Internet. Se estima que PHP es usado por cientos de miles de programadores y muchos millones de sitios informan que lo tienen instalado, sumando un gran por ciento de los dominios en Internet. . (Rodríguez. O, 2010).

### 1.14 Entorno de Desarrollo Integrado (IDE):

Un Entorno de Desarrollo Integrado es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones (Rodríguez. O, 2010).

El IDE de desarrollo seleccionado fue **PhpStorm** en su versión 7.1.3 ya que es un potente IDE especialmente diseñado a fin de proporcionar a los desarrolladores de HTML, JavaScript y PHP todas las herramientas necesarias para su trabajo. **PhpStorm** proporciona un editor de código enriquecido e inteligente para PHP con resaltado de sintaxis, configuración extendida de formateo del código, comprobación de errores sobre la marcha y finalización de código inteligente. (Softpedia, 2001-2014). El análisis completo del código fuente permite a **PhpStorm** proporcionar una finalización de código sofisticada incluso a un código no anotado, por ejemplo: el tipo de retorno de función se deduce desde su cuerpo e instrucciones de devolución, los tipos de propiedades de la clase (y declaraciones) se extraen desde el código de constructor. La aplicación viene con múltiples tipos de vistas (como jerarquía de llamada o método) con el fin de proporcionar a los usuarios rápida revisión del código.

Desarrolla pruebas de PHPUnit en **PhpStorm** y ejecútalas al instante desde el directorio, el archivo o la clase utilizando las opciones del menú contextual. Las configuraciones de ejecución/depuración dedicadas permiten que se ejecuten conjuntos necesarios de pruebas en cualquier momento posterior. Las pruebas ocurren en una dedicada interfaz de usuario de ejecución de pruebas mostrando los resultados generales y las estadísticas detalladas para la suite entera y cada prueba específica. **PhpStorm** ofrece numerosas opciones para depurar el código PHP que te permiten:

- Establecer puntos de interrupción condicionales con paso inteligente, lo que te permite seleccionar un método específico para ejecutar desde la cadena de llamadas.
- Inspeccionar variables locales pertinentes para el contexto y los relojes definidos por ti, incluyendo los arreglos de discos y los objetos complejos y editar los valores sobre la marcha.
- Evaluar una expresión en el tiempo de ejecución.
- Depurar una página en varias sesiones simultáneamente.
- Mantener una sesión de depuración mientras te desplazas entre páginas (Softpedia, 2001-2014).

**Estas son algunas de las características clave de "PhpStorm":**

- Editor inteligente:
- Editor de código PHP inteligente con finalización para la codificación más rápida.



- Análisis consciente de PHP:
- Conocimiento de código PHP avanzado y navegación rápida.
- Tu laboratorio de pruebas de código:
- Pruebas de unidad PHP asistidas con la interfaz de usuario de ejecutor de pruebas.
- Kit de herramientas de ajuste:
- Depurador gráfico integrado para el código PHP.. (Softpedia, 2001-2014)

### 1.15 Marcos de Trabajo:

Un marco de trabajo es una estructura de archivos y utilidades que aceleran la programación de una aplicación informática, proveyendo una metodología de trabajo que sistematiza y facilita la generación de formularios, funciones y módulos de uso común, permitiendo al desarrollador dedicar su atención hacia los aspectos específicos de cada aplicación. (Rodríguez. O, 2010)

Se seleccionó el marco de trabajo para el desarrollo del STI: Symfony

#### 1 Symfony.

Es un marco de trabajo para el desarrollo de aplicaciones web de forma rápida. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Symfony utiliza los siguientes componentes:

- **Propel:** ORM6 para el acceso a Base de Datos.
- **Creole:** Maneja la capa de abstracción de BD.
- **Phing:** Mapeador XML.
- **Pake:** Gestión de ejecución de Scripts.

A continuación se muestran algunas de sus características: (Potencier. F y otros, 2009).

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix).

- Independiente del sistema gestor de Bases de Datos.
- Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con bibliotecas desarrolladas por terceros. Usa el framework Propel para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases del modelo con todas las funcionalidades comunes de las entidades.

### 1.16 Gestor de Base de Datos:

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. Cualquier operación que el usuario hace contra la Base de Datos está controlada por el gestor.

El gestor almacena una descripción de datos en lo que llamamos diccionario de datos, así como los usuarios permitidos y los permisos. Tiene que haber un usuario administrador encargado de centralizar todas estas tareas.

Se determinó que el Gestor de Base de Datos (GBD) para almacenar toda la información generada por el STI fuera PostgreSQL.

PostgreSQL es un sistema de gestión de Bases de Datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. El director de este proyecto es el profesor Michael Stonebraker, y fue patrocinado por Defense Advanced Research Project Agency (DARPA), el

Army Research Office (ARO), el National Science Foundation (NSF). Este GBD es una derivación libre (OpenSource) de este proyecto, y utiliza el lenguaje SQL92/SQL99, así como otras características que se comentan más adelante. (Rodríguez. O, 2010)

### **Características de PostgreSQL.** (Rodríguez. O, 2010)

- ✓ **Atomicidad:** Asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- ✓ **Consistencia:** Asegura que sólo se empiece aquello que se puede acabar. Se ejecutan las operaciones que no van a romper las reglas y directrices de integridad de la Base de Datos.
- ✓ **Aislamiento:** Asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generarán ningún tipo de error.
- ✓ **Durabilidad:** Asegura que, una vez realizada la operación, esta persistirá y no se podrá deshacer.
- ✓ **Soporte de casi todos los sistemas operativos:** Linux, Unix, Mac OS, Beos, Windows, etc.
- ✓ **Documentación:** Posee una vasta documentación muy bien organizada, pública y libre.
- ✓ **Flexibilidad:** Altamente adaptable a las necesidades del cliente.
- ✓ **Soporte:** Posee soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python.

### **1.17 Herramienta CASE:**

Se propone utilizar como herramienta CASE Visual Paradigma en su versión 8.0 por ser una herramienta que sustenta UML y su licencia fue adquirida por la UCI. Es una herramienta que permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es considerada una herramienta muy completa, fácil de usar y con soporte multiplataforma. Su uso es sencillo para la creación de todo tipo de diagramas UML, para los que dispone de un número considerable de estereotipos, permitiendo mayor entendimiento de los mismos. La herramienta también proporciona abundante documentación de UML, como tutoriales, demostraciones interactivas y proyectos, facilitándole el uso a personas inexpertas (Serviño, 2013).

### 1.18 Servidor Web:

Un servidor Web consta de un intérprete HTTP6 el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Apache es un servidor web de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP. Se distribuye bajo una licencia especial Apache Software License. (Apache,2018)

Características de Apache:

- Es flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos.
- Es Multiplataforma.
- Es una tecnología gratuita de código fuente abierto.
- Es un servidor altamente configurable de diseño modular.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP.

(Apache,2018)

### Fundamentos de la selección:

Además de las características antes mencionadas, se escoge Apache como servidor Web, ya que es uno de los servidores web más utilizados, pues es una tecnología gratuita de código fuente abierto que ofrece instalaciones sencillas para sitios pequeños como es el caso y si se requiere se puede expandir. Es una solución sencilla, eficaz y rápida. Además, permite la creación de herramientas web dinámicas utilizando PHP que es el lenguaje de programación seleccionado.

### Conclusiones del capítulo:

En este capítulo se realizó un estudio de los STI, exponiendo en forma resumida sus principales características, así como su estructura y funcionamiento, llegando a la conclusión que para el sistema propuesto se hará uso de las Redes Neuronales como técnica de IA para su implementación. Además, en el estudio de las posibles metodologías, lenguajes y herramientas para el desarrollo del mismo, se concluyó que se utilizará como metodología de desarrollo AUP-UCI , UML como lenguaje de modelado, Visual Paradigma 8.0 como herramienta CASE, como lenguaje de programación del lado del cliente PHP y del lado del servidor HTML, CSS y JavaScript, como framework de apoyo Symfony,

además de IDE PhpStorm en su versión 7.1.3 , PostgreSQL como gestor de base de datos y Apache como servidor web.

## **CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL MÓDULO ESTUDIANTE PARA EL SISTEMA TUTORIAL INTELIGENTE.**

En el presente capítulo se describe la solución propuesta mediante la identificación de los requisitos funcionales y no funcionales con los que debe de cumplir el módulo. Para ello se hizo necesario seguir los pasos de la metodología propuesta en el capítulo anterior (AUP-UCI) y elaborar los artefactos fundamentales del proceso de desarrollo. Además, se presenta el diseño arquitectónico y se ejemplifica el empleo de los patrones del diseño, entre otros .

### **2.1 Propuesta de solución:**

Se propone desarrollar un sistema que clasifique los rasgos que se deben tener en cuenta al diagnosticar a un estudiante que vaya a cursar la asignatura Inteligencia Artificial 1 en la Universidad de las Ciencias Informáticas.

El módulo del alumno es el encargado de conocer las carencias del alumno, las debilidades y fortalezas a la hora de asimilar cierta información o contenido, debe reconocer las concepciones erróneas de estos. Está compuesto por dos submódulos: uno encargado de la información estática, el estilo de aprendizaje con que inicia el alumno el curso y otro dinámico que poseerá el conocimiento dado por la interacción entre el alumno y el sistema.

El módulo del alumno proporciona información necesaria para la retroalimentación, tiene estrecha vinculación con los contenidos y el método pedagógico en el módulo del tutor con los que se instruirán al estudiante así como con la interfaz en la que se presentarán dichos contenidos por el módulo del dominio.

Submódulo estático: se encarga de modelar el estilo de aprendizaje de los estudiantes a partir de los resultados del cuestionario índice de estilo de aprendizaje

Submódulo dinámico: se encarga de modelar el nivel de conocimiento previo que el estudiante posee antes de comenzar a interactuar con la plataforma, así como el que este adquiere como resultado de la interacción con el sistema.

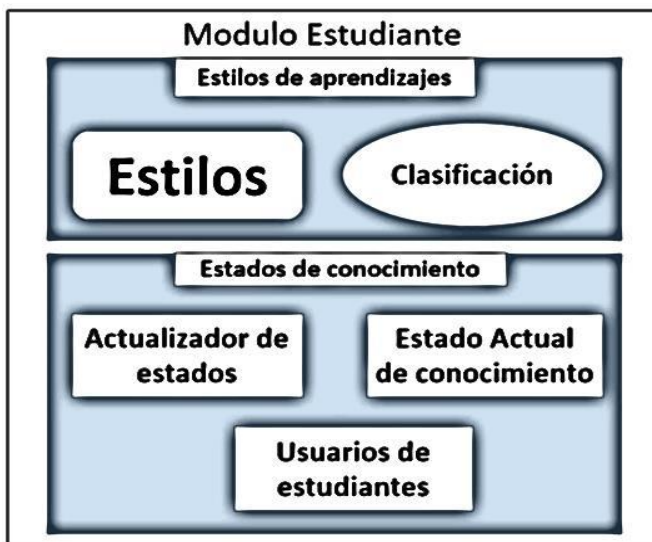


Figura.4 Propuesta de solución (Elaboración propia).

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. 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 acceder a un conocimiento determinado. En otras palabras, es una forma agrupar o 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 afectan al estudiante.

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 quien le enviará dichos resultados procesados.

## 2.2 Personas que interactúan con el sistema

La siguiente tabla muestra todas las personas que tienen acceso a la aplicación a desarrollar y la descripción de sus responsabilidades.

Tabla 1: Personas relacionadas con el sistema.

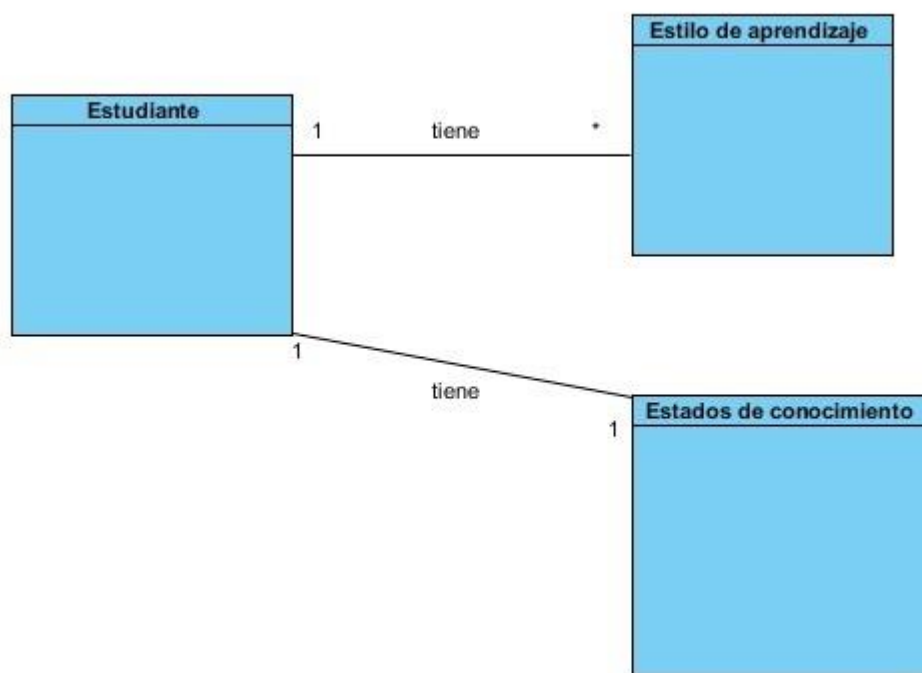
Personas relacionadas con el sistema	Responsabilidades
Usuario	Es cualquier persona que accede a la aplicación, se autentica y dependiendo del rol que desarrolle tendrá la posibilidad de realizar las acciones correspondientes a este
Estudiante	Es aquel usuario con un rol definido que se autentica y el sistema le brinda acceso a los diagnósticos para luego asignarle el estilo de aprendizaje que más se adecúa a este y pueda vencer las actividades programadas durante el curso de IA-1
Administrador	Encargado de implementar el Módulo Estudiante. Es la persona encargada de gestionar el sistema administrando las cuentas de los usuarios registrados en la aplicación y gestionando el contenido publicado en el mismo.

### 2.3 Modelo Conceptual:

El modelo conceptual se crea a partir del conjunto de descriptores e interrelaciones calificado como relevantes en la fase anterior. A través de ellos se puede explicar el estado y el funcionamiento de L'Albufera, siempre recordando que se trata de una modelización. El siguiente paso consiste en captar el estado y funcionamiento del sistema en una serie de indicadores, a través de cuyo seguimiento se puede evaluar el comportamiento y evolución del sistema en el tiempo. Los indicadores del modelo conceptual



ambiental de L´Albufera deben servir para evaluar el avance del sistema hacia un estado más sostenible. Un conjunto congruente de indicadores proporciona información sobre las tendencias que resulten de las actuaciones y medidas de gestión que se realizan en el ámbito su influencia. (Sommerville, 2010).



**Figura 5: Modelo Conceptual (elaboración propia).**

**Estudiante:** Usuario con un rol definido al cual se le asigna un estilo de aprendizaje y va actualizando su estado de conocimiento

**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.

**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

## 2.4 Obtención de requisitos:

Según la IEEE un requisito es:

1. Una condición o capacidad requerida por un usuario para resolver un problema o alcanzar un objetivo.
2. Una condición o capacidad que debe cumplir o poseer un sistema o componente de sistema para satisfacer un contrato, estándar, especificación, o cualquier otro documento impuesto formalmente.
3. Una representación documentada de una condición o capacidad de lo explicado en los puntos 1 o 2

*(IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society Press. 1990).*

La especificación de **requisitos de software** (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software.

### 2.4.1 Técnicas de levantamientos de los requisitos de software:

Una etapa fundamental en proyectos de ingeniería de software, es la identificación y documentación de los requerimientos del futuro sistema al comienzo del proyecto, pues en numerosas ocasiones se ha demostrado que es cuando pueden prevenirse errores que puedan significar el fracaso del proyecto. En la Ingeniería de requisitos, el levantamiento de requerimientos se refiere a la identificación y documentación de los requerimientos de un sistema, a partir de los usuarios, clientes o interesados (Stakeholders). A la práctica también se le conoce como Recopilación de requerimientos. (Sommerville, 2010).

#### - Análisis de documentación

Consiste en obtener la información sobre los requerimientos funcionales y requerimientos no funcionales de software a partir de documentos que ya están elaborados. Es útil cuando los expertos en la materia no están disponibles para ser entrevistados o ya no forman parte de la organización. Utiliza la documentación que sea relevante al requerimiento que se está levantando.

Durante la investigación se analizaron varios documentos de sistemas con estrecha relación con la solución a desarrollar que permitieron obtener los requisitos del sistema con mayor claridad.

### - Entrevistas

Se realizan con los usuarios o interesados clave. Direccionan al usuario hacia aspectos específicos del requerimiento a levantar. Son útiles para obtener y documentar información detallada sobre los requerimientos y sus niveles de granularidad. Pueden ser entrevistas formales o informales. Una clave es mantenerse enfocado en los objetivos de la entrevista. Las preguntas abiertas son útiles para identificar información faltante. El éxito de las entrevistas depende del grado de conocimiento del entrevistador y entrevistado, disposición del entrevistado de suministrar información, buena documentación de la discusión y en definitiva de una buena relación entre las partes.

Se realizaron variadas entrevistas con el cliente y con el equipo técnico tratando de comprender y tener en cuenta cada detalle, gracias a esto se lograron obtener distintos requisitos que debe cumplir el sistema a implementar.

### 2.5 Requisitos Funcionales:

Un **requisito funcional** define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone un sistema debe cumplir. Los requisitos de comportamiento para cada requisito funcional se muestran en los casos de uso. Son complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación. (Wieggers, Karl E. (2003). Software Requirements 2)

A partir de lo planteado anteriormente se definieron los siguientes requisitos funcionales:

**RF1: Autenticar usuario.**

**RF2: Insertar estudiante**

**RF3: Eliminar estudiantes**

**RF4: Modificar estudiantes**

**RF5: Buscar estudiantes.**

RF6: Mostrar Formulario.

RF7: Clasificar estado de conocimiento.

RF8: Clasificar estilos de aprendizaje

Tabla 2: Descripción de los requisitos funcionales.

No.	Requisito Funcional	Descripción	Complejidad	Prioridad para el cliente
1.	Autenticar usuario	La aplicación deberá permitir al usuario autenticarse utilizando usuario y contraseña para acceder al sistema.	Alta	Alta
2.	Insertar Estudiante	El sistema debe permitir insertar un estudiante	Alta	Alta
3.	Eliminar estudiante	El sistema debe permitir eliminar un estudiante seleccionado	Alta	Alta
4.	Modificar estudiantes	El sistema debe permitir modificar un estudiante seleccionado	Alta	Alta
5.	Buscar estudiantes	El sistema debe permitir buscar un estudiante	Alta	Alta

6.	Mostrar Formulario	El sistema debe mostrar al estudiante una vez que se autentique el formulario a rellenar		
7.	Clasificar Estilo de Aprendizaje	El sistema debe permitir asignarle al estudiante después de haber dado respuesta a todas las encuestas del formulario el estilo de aprendizaje que más se adecue a este .	Alta	Alta
8.	Clasificar el estado del conocimiento	El sistema debe clasificar el estado de conociendo del estudiante actual y ser capaz de ir actualizándolo según los avances del estudiante	Alta	Alta

## 2.6 Requisito no funcional:

Un **requisito no funcional** o **atributo de calidad** es, en la ingeniería de sistemas y la ingeniería de software, un requisito que sabe bien y especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento, por eso suelen denominarse Atributos de calidad de un sistema. Queda entonces el requisito no funcional, que son las restricciones o condiciones que impone el cliente al programa que necesita, por ejemplo, el tiempo de entrega del programa, el lenguaje o la cantidad de usuarios (Stellman, Andrew; Greene, Jennifer (2005)

### Confiabilidad

RNF1- La aplicación debe ser capaz de manejar los errores y recuperarse.

### Usabilidad

RNF2- Memoria RAM del dispositivo con capacidad mínima de 512 Mb.

RNF3- Almacenamiento interno con capacidad mínima de 1024 Mb.

RNF4- La aplicación deberá soportar los idiomas inglés y español.

RNF5- Soporte para conexiones WIFI.

### Interfaz

RNF6- La aplicación tendrá una interfaz de usuario sencilla e intuitiva.

### Portabilidad

RNF9- La aplicación deberá funcionar correctamente en diferentes sistemas operativos.

## 2.7 Diagrama de Casos de Uso del Sistema:

Un caso de uso es una secuencia de transacciones que son desarrolladas por un **sistema** en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas o lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo, la relación y la generalización son relaciones. (Gutiérrez, J. 2008.)

**Tabla3: Descripción de actores.**

Actor	Descripción
Administrador	Encargado de gestionar cada usuario que interactúa con el sistema
Estudiante	El estudiante debe autenticarse para tener acceso al sistema desarrollar todas las actividades académicas destinadas para él y poder llevar un registro de su avance

	académico.
Sistema	Muestra los contenidos específicos dependiendo del rol que desempeñe cada usuario.

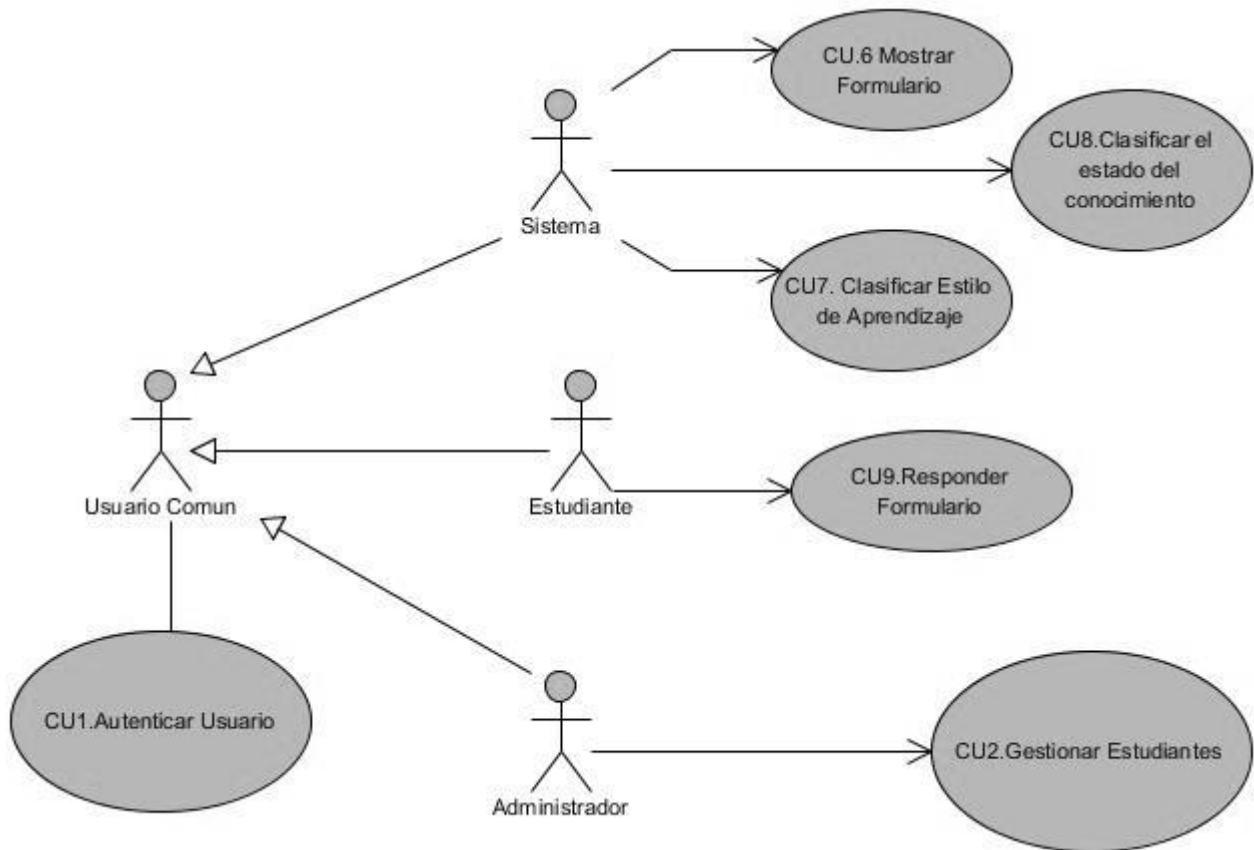


Figura 6: Casos de uso del sistema (elaboración propia).

## 2.8 Modelo de datos:

El modelo Entidad-Relación facilita al ingeniero del software identificar los objetos de datos y sus relaciones por medio de una notación gráfica, define en su análisis estructural los datos que son adicionados, almacenados, modificados y producidos dentro de una aplicación (Pressman, 2005).

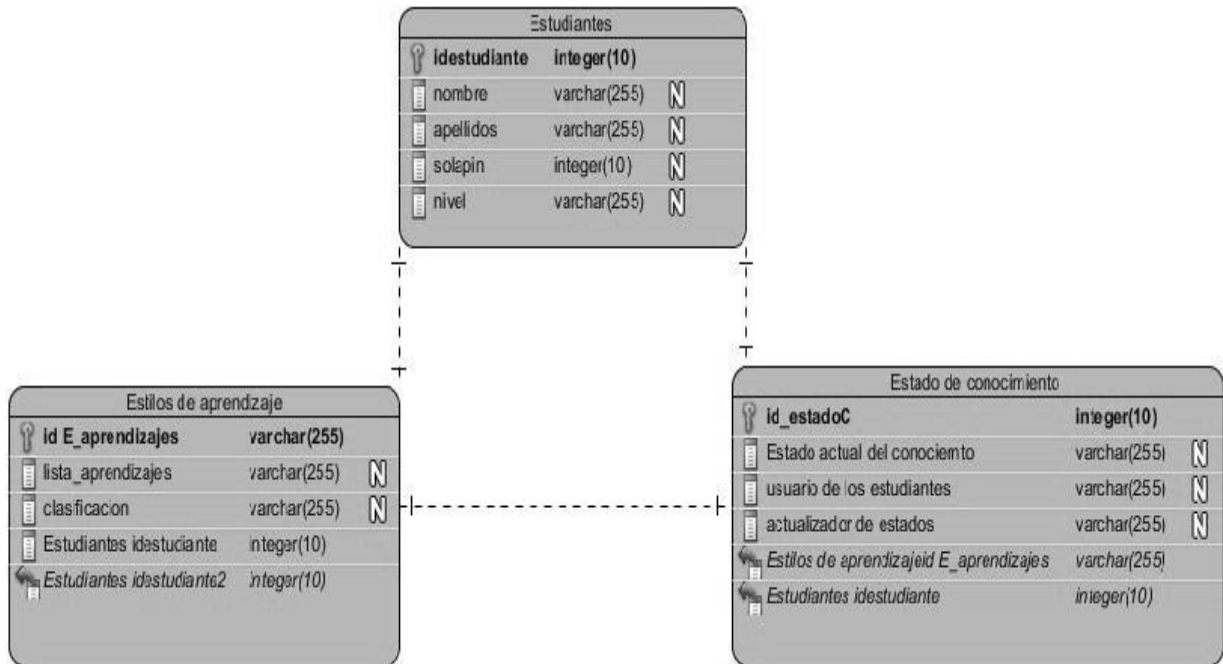


Figura 7: Modelo de datos (elaboración propia).

## 2.9 Arquitectura:

Los patrones de diseño son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Para el desarrollo del STI el patrón que se utilizó es el Modelo-Vista - Controlador (MVC), ya que *Symfony* el *framework* seleccionado se basa en este patrón.

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde el modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos, la vista transforma esta información obtenida por el modelo en las páginas web a las que acceden los usuarios y el controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista. (Pedro Veloso Hernández,2009).



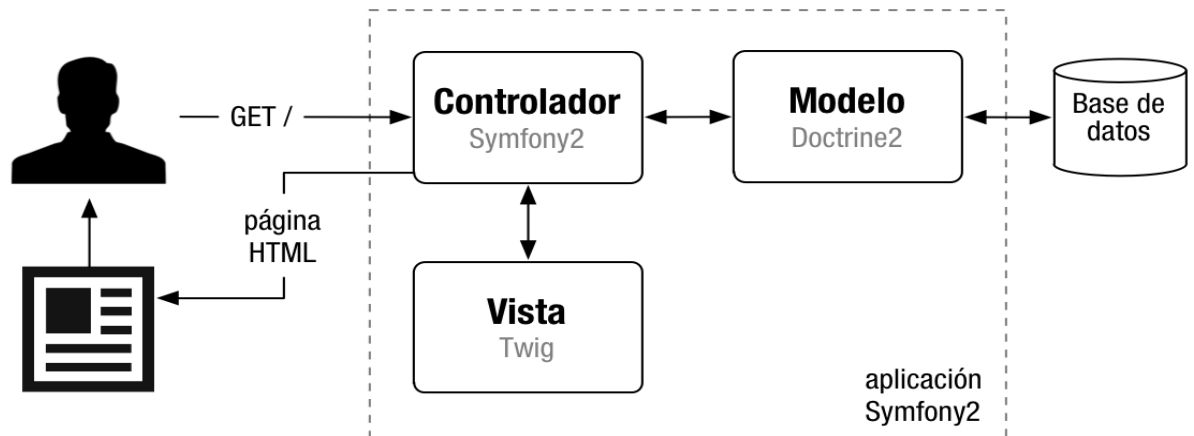


Figura 8: Esquema simplificado de la arquitectura MVC de Symfony2.

**Modelo:** Es todo lo referente a la gestión de la información y la interacción con los datos de nuestra aplicación (comúnmente bases de datos) este modelo realizará acceso a dicha información como también podrá realizar actualizaciones y depuraciones de los datos gestionados. Toda petición de acceso a la información siempre pasará por esta capa.

**Controlador:** Este es el puente entre la vista y el modelo ya que el usuario solicitará información mediante la vista y este pasará por el controlador para posteriormente realizar la petición al modelo, habitualmente es llamado la capa de lógica del negocio.

**Vista:** Esta capa nos mostrará la información formateada y ordenada, es el resultado de todo lo que el modelo interaccione con los datos, este lo muestra mediante la interfaz de usuario, habitualmente llamado la capa de presentación.

## 2.10 Patrones de Diseño:

Los patrones de diseño se han convertido en una técnica importante para el reúso del conocimiento de software. Cada patrón provee información sobre su diseño, describiendo las clases, métodos y relaciones que resuelven un problema de diseño en particular, estos han sido agrupados y organizados en catálogos cada uno dando diferentes clasificaciones y descripciones (CAMACHO, y otros, 2004).

### 2.11 Patrones para asignar responsabilidades:

En diseño orientado a objetos, **GRASP** son patrones generales de software para asignación de responsabilidades, es el acrónimo de "GRASP (object-oriented design General Responsibility Assignment Software Patterns)". Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software (RODRIGUEZ, 2014).

#### **Experto:**

El GRASP de experto en información es el principio básico de asignación de responsabilidades. Nos indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento) (CAMACHO, y otros, 2004).

#### **Creador:**

El patrón creador nos ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que:

- Tiene la información necesaria para realizar la creación del objeto, o
- Usa directamente las instancias creadas del objeto, o
- Almacena o maneja varias instancias de la clase
- Contiene o agrega la clase.

Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización (RODRIGUEZ, 2014).

### **Controlador:**

El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento (Bermudez, 2013).

### **Alta Cohesión:**

El patrón es el encargado de asignar una responsabilidad de modo que la cohesión siga siendo alta, con el objetivo de que cada una de las clases tenga responsabilidades específicas, por lo que cada clase tiene determinadas responsabilidades y colabora con otras para llevarlas a cabo, como por ejemplo las clases controladoras modifican los datos haciendo uso de las clases modelos, las cuales solo tienen la responsabilidad de hacer cambios en la base de datos. (Escalante, 2016).

### **Bajo Acoplamiento:**

Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases (CAMACHO, y otros, 2004).

### **2.12 Patrones GOF:**

Según (Valdés, 2014) los patrones GoF describe soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos, permiten enfrentarse a la programación de software propiciando reutilización y extensibilidad de soluciones que han funcionado en el pasado para garantizar la calidad de los mismo, estos patrones son conocidos como los patrones de la pandilla de los cuatro (GoF, gang of four). El objetivo principal de los patrones es facilitar la reutilización de diseños y arquitecturas software que han tenido éxito capturando la experiencia y haciéndola accesible a los no expertos.

**Instancia única (Singleton):** Este patrón garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Este tipo de patrón es muy utilizado en frameworks como AngularJS, Symfony entre otros. Se evidencia en la clase Controller, la cual define varias variables globales de instancia única que permiten la comunicación con el modelo.

### 2.13 Diagrama de clase del diseño:

El Diagrama de Clases de Diseño describe gráficamente las especificaciones de las Clases de Software y las Interfaces en una aplicación. Es una representación concreta de lo que se debe implementar. Estos diagramas representan la parte estática del sistema a través de la representación de las clases y sus relaciones (Pressman,2010).

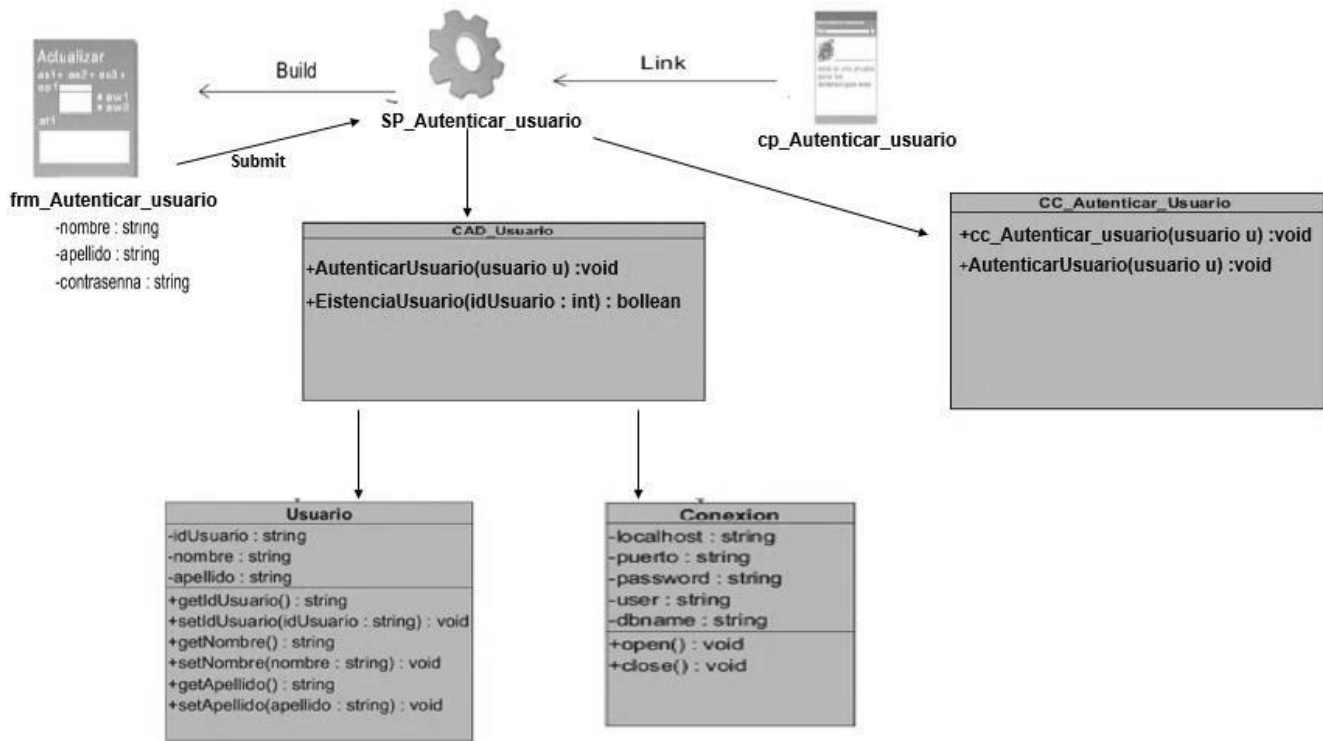


Figura 9: Diagrama de clases Registrar Usuario (elaboración propia)

### 2.14 Conclusiones del capítulo:

- La propuesta de solución descrita brinda las características fundamentales del sistema a desarrollar sirviendo de guía para la implementación del mismo.
- Mediante las diferentes técnicas de obtención de requisitos utilizadas resultaron 8 requisitos funcionales los cuales definen la función del sistema de software y sus componentes, además 9 requisitos no funcionales los cuales precisan las condiciones o restricciones impuestas por el cliente para lograr el sistema deseado.
- La descripción de casos de uso del sistema, y los diagramas de clases de diseño servirán de guía para la implementación del módulo y formarán parte su documentación técnica, lo cual facilitará futuras labores de mantenimiento.

- La elección del patrón arquitectónico Modelo Vista Controlador y la aplicación de patrones de diseño, evidenciará la utilización de buenas prácticas durante el desarrollo del módulo, lo cual repercute positivamente en la calidad final del mismo.

## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO ESTUDIANTE PARA EL SISTEMA TUTORIAL INTELIGENTE.

En el presente capítulo se analizan los aspectos relacionados con la implementación y pruebas de la solución propuesta para el desarrollo del módulo estudiantes para el sistema tutorial inteligente como los estándares de codificación del lenguaje seleccionado. Se presenta la estructuración del mismo a través del diagrama de componentes, su distribución física mediante el diagrama de despliegue, se define y realiza el proceso de aplicación de las pruebas para comprobar el correcto funcionamiento del módulo con el objetivo de detectar y enmendar la mayor cantidad de errores posibles en la implementación de los requisitos funcionales del módulo.

### 3.2 Estándares de codificación

Los estándares de codificación son muy importantes en cualquier aplicación, ya que permiten que el código sea uniforme y fácil de mantener. El mismo comprende los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (Muñoz, 2016).

Además, si se aplica de forma continua un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas y posteriormente, se efectúan revisiones del código de rutinas, existen posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener (Muñoz, 2016).

#### Visión general

- El código DEBE seguir el estándar PSR-1.
- El código DEBE usar 4 espacios como indentación, no tabuladores.
- NO DEBE haber un límite estricto en la longitud de la línea; el límite DEBE estar en 120 caracteres; las líneas DEBERÍAN tener 80 caracteres o menos.
- DEBE haber una línea en blanco después de la declaración del namespace, y DEBE haber una línea en blanco después del bloque de declaraciones use.
- Las llaves de apertura de las clases DEBEN ir en la línea siguiente y las llaves de cierre DEBEN ir en la línea siguiente al cuerpo de la clase.

- Las llaves de apertura de los métodos DEBEN ir en la línea siguiente y las llaves de cierre DEBEN ir en la línea siguiente al cuerpo del método.
- La visibilidad DEBE estar declarada en todas las propiedades y métodos; abstract y final DEBEN estar declaradas antes de la visibilidad; static DEBE estar declarada después de la visibilidad.
- Las palabras clave de las estructuras de control DEBEN tener un espacio después de ellas, las llamadas a los métodos y las funciones NO DEBEN tenerlo.
- Las llaves de apertura de las estructuras de control DEBEN estar en la misma línea y las de cierre DEBEN ir en la línea siguiente al cuerpo.
- Los paréntesis de apertura en las estructuras de control NO DEBEN tener un espacio después de ellos y los paréntesis de cierre NO DEBEN tener un espacio antes de ellos.

Este ejemplo incluye algunas de las siguientes reglas a modo de visión general:

```

<?php
namespace Proveedor\Paquete;

use FooInterfaz;
use BarClase as Bar;
use OtroProveedor\OtroPaquete\BazClase;

class Foo extends Bar implements FooInterfaz
{
    public function funcionDeEjemplo($a, $b = null)
    {
        if ($a === $b) {
            bar();
        } elseif ($a > $b) {
            $foo->bar($arg1);
        } else {
            BazClase::bar($arg2, $arg3);
        }
    }

    final public static function bar()
    {
        // cuerpo del método
    }
}
    
```

Figura 10: Ejemplo de reglas a modo de visión general (Lenguaje PHP).



### 3.3 Diagrama de componentes

Dentro del modelo de implementación se encuentran los diagramas de componentes. Un diagrama de componentes representa cómo un sistema de software es dividido en componentes, mostrando las dependencias que existen entre ellos. Los componentes físicos incluyen archivos, bibliotecas compartidas, módulos, ejecutables y paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema; estos son utilizados para modelar la vista estática y dinámica de un sistema y; muestran la organización y las dependencias entre un conjunto de componentes. (Jacobson, y otros, 2000).

A continuación, se muestra el diagrama de componentes del módulo estudiante. Donde el paquete Vista contiene las clases referentes a la interfaz de usuario y que interactúan con el mismo, el paquete Controlador contiene las clases principales que gestionan la lógica, el paquete Modelo contiene el componente persistente.

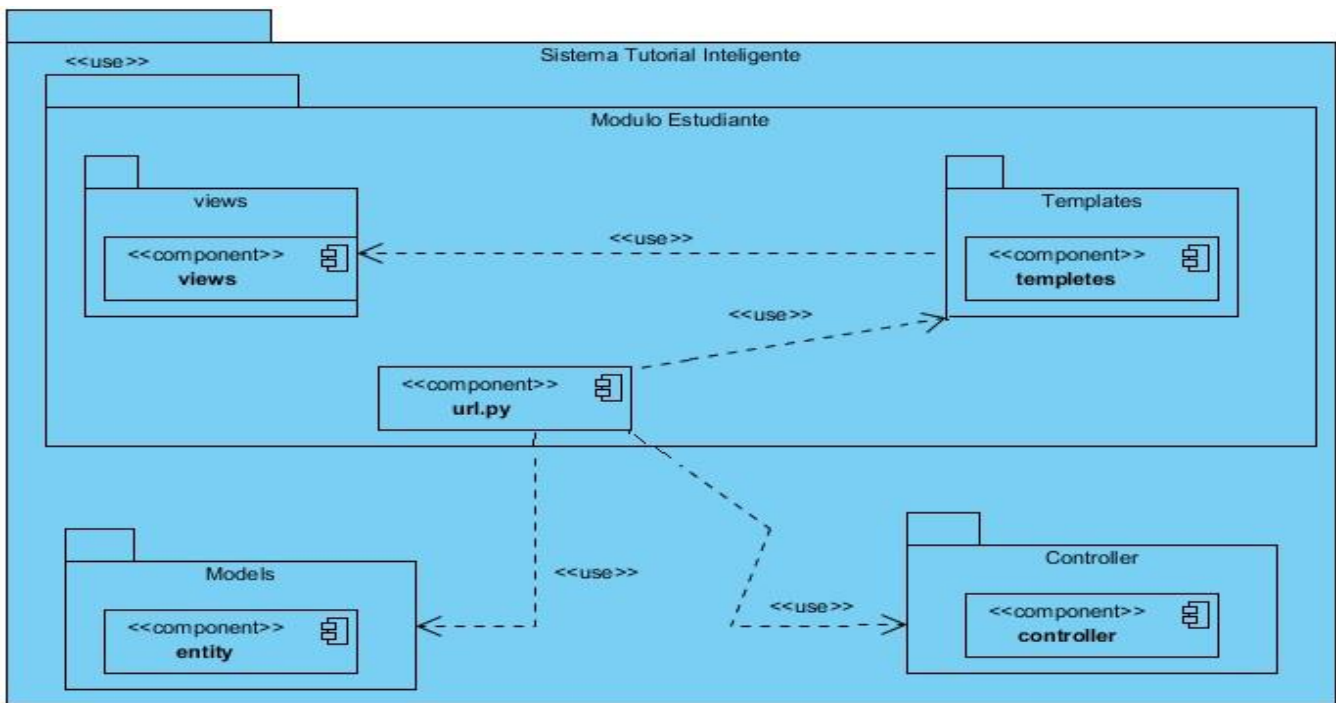


Figura 11: Diagrama de Componentes (elaboración propia).

### 3.4 Diagrama de Despliegue

El diagrama de despliegue es un tipo de diagrama del UML que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones. Muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Un nodo es un recurso de ejecución como un computador, dispositivo, conectados por asociaciones de comunicación tales como enlaces de red, conexiones HTTP, etc. (*Lenguaje-Unificado-de-Modelado-UML* , 2010).

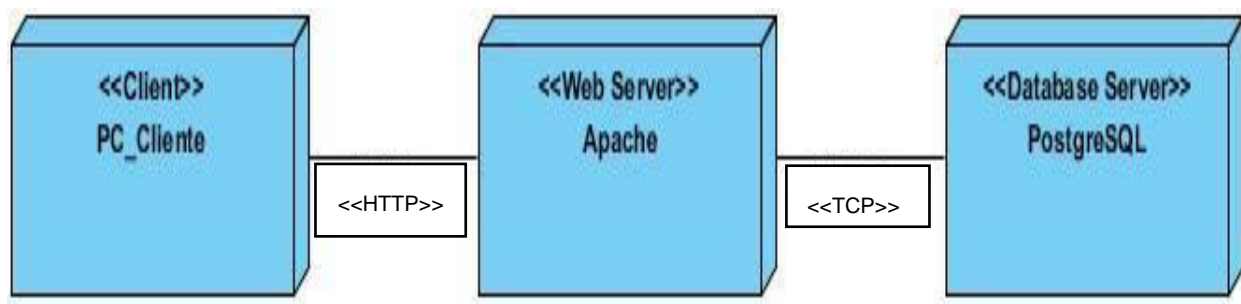


Figura 12: Diagrama de despliegue. (elaboración propia)

Descripción de los componentes:

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

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

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

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

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

### 3.5 Pruebas a realizar:

En esta etapa se propone explicar mejor la estrategia de prueba al módulo:

**Prueba de Integración:** Las Pruebas de Integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño. (Pressman, 2010)

La integración del sistema implica identificar grupos de componentes que proporcionan alguna funcionalidad del sistema e integrar estos añadiendo código para hacer que funcionen conjuntamente. Algunas veces, primero se desarrolla el esqueleto del sistema en su totalidad y se le añaden los componentes. Esto se denomina integración descendente. (Somerville, 2009)

De forma alternativa, pueden integrarse primero los componentes de infraestructura que proporcionan servicios comunes, tales como acceso a la base de datos y redes, a continuación, pueden añadirse los componentes funcionales. Esta es la integración ascendente. (Somerville, 2009) Dicho método es el utilizado en la integración del módulo.

**Prueba de Caja Blanca:** La prueba de Caja Blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes o código para derivar casos de prueba. Al usar los métodos de prueba de Caja Blanca, puede derivar casos de prueba que: garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, revisen todas las decisiones lógicas en sus lados verdadero y falso, ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y revisen estructuras de datos internas para garantizar su validez. (Pressman, 2010)

**Prueba de Aceptación:** Cuando se construye un software a la medida para un cliente, se realiza una serie de Pruebas de Aceptación a fin de permitir al cliente validar todos los requerimientos. Realizada por el usuario final en lugar de por los ingenieros de software, una Prueba de Aceptación puede variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, la Prueba de Aceptación puede realizarse durante un período de semanas o meses, y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema (Pressman, 2010).

### 3.6 Conclusiones del capítulo

- Se especificó el uso de los estándares de codificación para lograr un estilo claro y organizado del código durante la fase de implementación.
- Se realizó el diagrama de componentes, con el propósito de mostrar los componentes del sistema y sus relaciones.
- La construcción del modelo de despliegue ofrece una distribución física del sistema mostrando los principales requisitos del sistema.
- La estrategia de pruebas desarrollada permitirá detectar y corregir los errores de la etapa de implementación, lo cual aumenta la calidad del producto.

### GENERALES

#### CONCLUSIONES GENERALES

Luego de finalizada la investigación referida a la implementación del módulo estudiante del Sistema Tutorial Inteligente para la asignatura de Inteligencia Artificial 1 se concluye:

- Mediante el estudio teórico relacionados a los STI, así como su estructura y funcionamiento , se propone el uso de las Redes Neuronales como técnica de IA para su implementación y AUP-UCI como metodología de desarrollo.
- La descripción de casos de uso del sistema, y los diagramas de clases de diseño servirán de guía para la implementación del módulo y formarán parte su documentación técnica, lo cual facilitará futuras labores de mantenimiento.
- Mediante las diferentes técnicas de requisitos utilizadas resultaron 8 requisitos funcionales los cuales definen la función del sistema de software y sus componentes, además 9 requisitos no funcionales los cuales precisan las condiciones o restricciones impuestas por el cliente para lograr el sistema deseado.
- La investigación sobre las ventajas y desventajas de las herramientas propuestas confirmó que eran las indicadas para el desarrollo del trabajo.
- Se propusieron diferentes pruebas de calidad para aplicar al sistema una vez implementado y asegurar el adecuado funcionamiento del producto final

## RECOMENDACIONES

Una vez concluida la investigación, para la futura implantación del sistema el autor recomienda:

- Desarrollar sistemas similares para otras asignaturas esenciales en el proceso de formación de profesionales de la ingeniería informática.

## REFERENCIAS BIBLIOGRÁFICAS

- **Nobrega, Maria de. 2005.** Curso de Ssistemas de Informacion II. [En línea] 2005. [https://curso\\_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php..](https://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php..)
- **Pérez López, Cesar. 2000.** *Técnicas de muestreo estadístico: teoría, práctica y aplicaciones informáticas.* s.l. : Alfaomega Grupo Editor, 2000.
- **Perez, J.E. 2007.** *Introducción a CSS.* 2007.
- **Pressman, Roger. 2010.** *Ingeniería de Software. Un enfoque Práctico.* 2010.
- **Real Academia Española. 2014.** *Diccionario de la lengua española.* Madrid : Espasa, 2014.
- **Rodríguez, Fran Gil. 2012.** *Experto en Drupal 7 Nivel Inicial.* 2012.
- **Sánchez, Tamara Rodríguez. 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI.* Habana : s.n., 2015.
- **Simon, Herbert Alexander. 1982.** *The New Science of Management Desicion.* Pittsburgh : s.n., 1982.
- **Skvorc, Bruno. 2015.** Best PHP IDE in 2015 - Survey Results. [En línea] 2015. [Citado el: 15 de noviembre de 2015.] <https://www.sitepoint.com/best-php-ide-2014-survey-results/>.
- **Somerville, Ian. 2009.** *Ingeniería del Software Séptima Edición.* 2009.
- **UML, Lenguaje. 2018.** *Lenguaje unificado del modelado.* 2018.
- **VANLEHN. 1998.** *sistemas tutores inteligentes orientados a la enseñanza.* Buenos Aires : Facultad Regional Buenos Aires Universidad Tecnológica Nacional., 1998. 1135-9250.
- **Wenger. 1997.** *sistemas tutores inteligentes orientados a la enseñanza.* 1997.
- **phpMyAdmin. 2016.** *phpMyAdmin.* 2016. <https://www.phpmyadmin.net>
- **Wolf. 1984.** *SISTEMAS TUTORES INTELIGENTES ORIENTADOS A LA enseñanza.* Buenos Aires : s.n., 1984. 1135-9250.
- **Zulma Cataldi. 2009.** *Articulos sobre la Arquitectura de un Sistem Tutorial Inteligente.* España : s.n., 2009.
- **Bedoya, Chávez. 2017.** *Diagramas de Clase UML.* 2017.
- **Bell. 2004.** *Diagrama de Componentes.* 2004.
- **Bermudez, Camilo. 2013.** *Patrones de Diseño.* 2013

- VALERO, Carmen Cantillo; RODONDO, Margarita Ruora; PALACÍN, Ana Sánchez. Tendencias actuales en el uso de dispositivos móviles en educación. *La Educación digital magazine*, 2012, vol. 147, p. 1-21.
- RAMOS, Elme Carballo; PANADEIROS, Antonio Felipe; SÁNCHEZ, Mario Ares. *Centro de gestión de la información y el conocimiento (CEGIC), como ente dinamizador (...)*. En: *Memorias del programa científico Universidad 2010*. Editorial Universitaria, 2010.
- GRANMA. La informatización de la sociedad, una prioridad para Cuba, 2014
- UCI. Universidad de las Ciencias Informáticas. 2013. [http:// www.uci.cu](http://www.uci.cu).
- SANTILLÁN, Arturo García; CHÁVEZ, Milka Elena Escalera; NAVARRO, Rubén Edel. Variables asociadas con el uso de las TIC como estrategia didáctica en el proceso enseñanza-aprendizaje de la matemática financiera: Una Experiencia desde el aula de clase. *Revista Iberoamericana de Evaluación Educativa*, 2011, vol. 4, no 2, p. 118-135.
- CARMENANTE, Fabián Ramírez. *Revista Informática Científica*, 2015.
- SANTANA, Eunises Cabezas. Una mirada al proceso de enseñanza-aprendizaje de la construcción de textos escritos en la formación inicial de profesores de idioma inglés. *Atenas*, 2012, vol. 3, no 19, p. 48-58.
- ROJANO RAMOS, Santiago; LÓPEZ GUERRERO, María del Mar; LÓPEZ GUERRERO, Gema. Desarrollo de tecnologías de la información y la comunicación para reforzar los procesos de enseñanza y aprendizaje en ciencias en el grado de maestro/a en educación infantil de la Universidad de Málaga. *Educación química*, 2016, vol. 27, no 3, p. 226-232.
- MELCHOR, Alex. Prezi. *Prezi*. [En línea] 22 de noviembre de 2012. [Citado el: 23 de marzo de 2019.] Consultado en: [https://prezi.com/e\\_gpb7xev\\_im/tutorial-diagramas-de-despliegue/](https://prezi.com/e_gpb7xev_im/tutorial-diagramas-de-despliegue/).
- ABRAHAMSSON, Pekka, et al. Mobile-D: An Agile Approach for Mobile Application Development. *arXiv preprint arXiv:1709.06820*, 2017.
- GRAHAM, Dorothy; VAN VEENENDAAL, Erik; EVANS, Isabel. *Foundations of software testing: ISTQB certification*. Cengage Learning EMEA, 2008.
- KANAT-ALEXANDER, Max. *Code Simplicity: The Science of Software Design*. O'Reilly & Associates, 2012.
- Docker, Inc. Docker: Enterprise Application Container Platform. [En línea] 2019. Consultado en: <https://www.docker.com/>.



- **Rodríguez., Victor Angel Fong Rios y Roger Armando González. 2013.** *Sistema de gestión de información para los resultados de las pruebas de eficiencia física en la Universidad de las Ciencias Informáticas.* Universidad de las Ciencias Informáticas, La Habana: s.n., 2013.
- **Fuentes., Osay González. 2013.** *Curso para Principiantes en Symfony 2-- Parte 1. La Comunidad de la UCI de PHP.* [En línea] 31 de mayo de 2013. [https://php.uci.cu/articles.php?article\\_id=87](https://php.uci.cu/articles.php?article_id=87). 2013.
- **Softpedia.(2001-2014).** *PhpStorm.* Obtenido de PhpStorm: <http://www.softpedia.es/programa-PhpStorm-161476.html>
- **Canós, J. H.** *Métodologías Ágiles en el Desarrollo de Software.* Valencia : Universidad Politécnica de Valencia.
- **Maura Berta Hidalgo Rosales, M. d. (2014).** HERRAMIENTA PARA GESTIONAR EL CONOCIMIENTO : HERRAMIENTA PARA GESTIONAR EL CONOCIMIENTO EN CENTROS DE AUTO-APRENDIZAJE DE IDIOMAS EXTRANJEROS.
- **Larman, C. (s.f.).** *UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado.* Prentice Hall.
- **Bergmann, S. (2014).** *PHPUnit Manual.* Recuperado el 03 de 04 de 2014, de <http://phpunit.de/manual/current/en/phpunit-book.pdf>
- **Pacheco, N. (2011).** *Manual de Symfony2.*

## **Anexos**

Anexo 1:

Preguntas de la entrevista realizada a los clientes, que en este caso son los profesores de IA:

Pregunta 1:

¿Cuáles son las principales herramientas y metodologías utilizadas para el posterior desarrollo de el módulo estudiante del sistema tutorial inteligente?

Pregunta 2:

¿Cuáles son las principales deficiencias y dificultades que presenta los estudiantes de la UCI en la asignatura inteligencia artificial 1?

Pregunta 3:

¿Qué aportes y beneficios traerá la futura implantación del módulo estudiante de un sistema tutorial inteligente en la enseñanza de la IA 1 en la UCI?

**Anexo 2: Casos de uso del Sistema:**

CU_1	Autenticar usuario
Actor	Usuario
Descripción	El usuario debe autenticarse con sus datos y dependiendo de su rol en el Sistema le otorgan los permisos que le correspondan

*Tabla 4:CU1: Autenticar usuario*

CU_2	Gestionar Estudiantes
Actor	Administrator
Descripción	El administrador gestiona cada estudiante que interactúa con el sistema .

*Tabla 5:CU2: Gestionar Estudiantes*

CU_6	Mostrar Formulario
Actor	Sistema
Descripción	El Sistema debe mostrar el formulario a los estudiantes posterior a que se autentiquen .

Tabla 6:CU6: Mostrar Formulario

CU_7	Clasificar estado de conocimiento
Actor	Sistema
Descripción	El Sistema debe ir clasificando el conocimiento del estudiante durante su trayectoria en las actividades del curso .

Tabla 7:CU7: Clasificar estado de conocimiento

CU_8	Clasificar estilo de aprendizaje
Actor	Sistema
Descripción	El Sistema debe clasificar el estilo de aprendizaje que se adecue mas a cada estudiante

Tabla 8:CU8: Clasificar estilo de aprendizaje