

Universidad de las Ciencias Informáticas

Facultad 3



**Título: Módulo Perfil de Usuario del Tutor Virtual
para la Evaluación del Aprendizaje Autónomo de
Idiomas.**

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

Autor

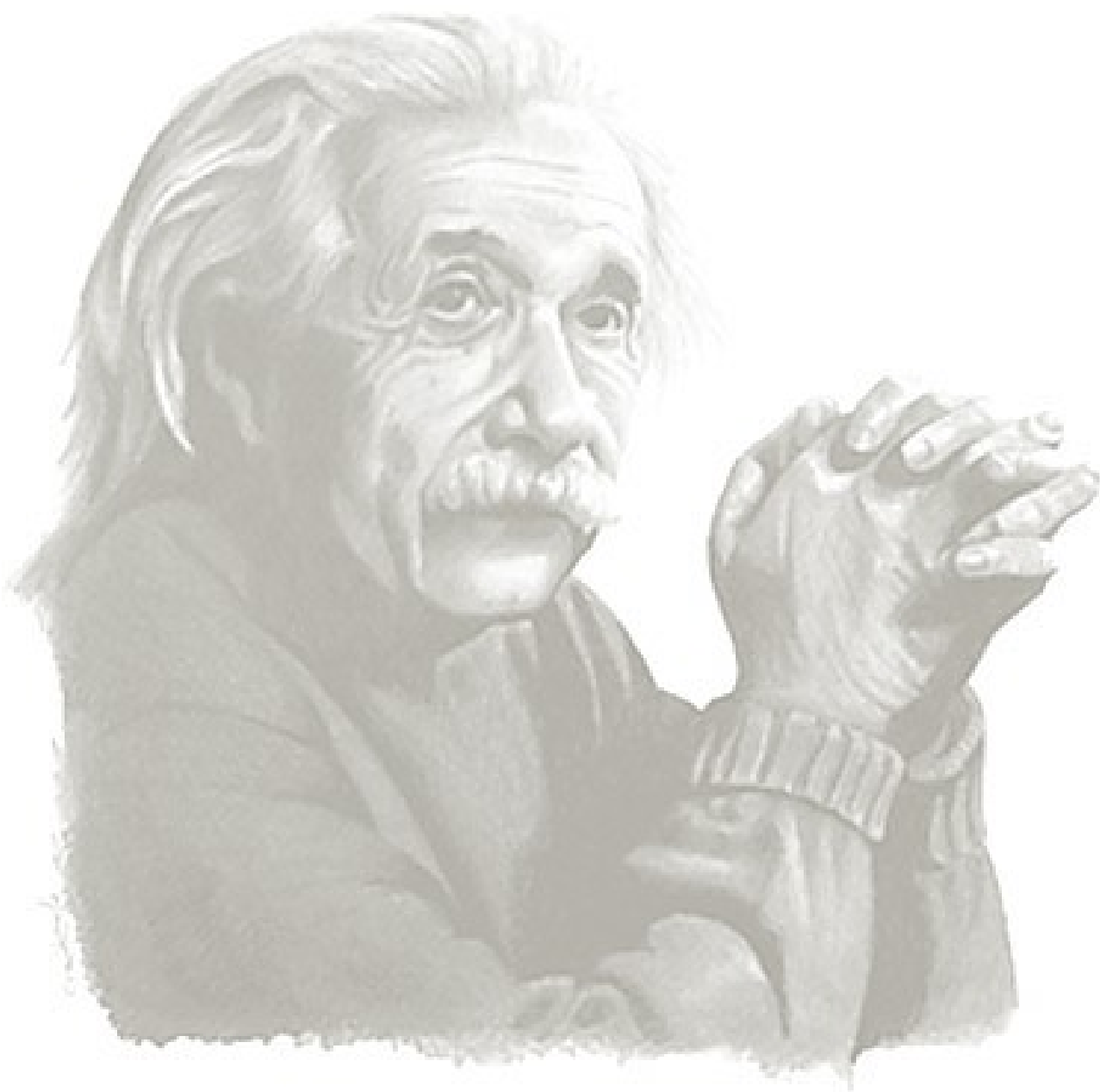
Adrián González Ruiz

Tutores

MSc. Yoan Martínez Márquez.

Ing. Sándor Rodríguez Prieto.

La Habana, Junio 2012.



"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."

Albert Einstein.

DECLARACIÓN DE AUTORÍA

Declaración de autoría.

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Adrián González Ruiz

Ms C. Yoan Martínez Márquez

Firma del Autor

Firma del Tutor

Ing. Sándor Rodríguez Prieto

Firma del Tutor

AGRADECIMIENTOS

Agradecimientos.

Me siento profundamente agradecido con todas las personas que se han cruzado en mi vida y que me han inspirado, conmovido e iluminado con su presencia.

En primer lugar agradecer a mi familia en general, por ser constante veladora de mis largos años de estudios.

En especial a mis abuelos, por siempre estar a mi lado y ser mi refugio más seguro. Ustedes son las joyas más valiosas de mi vida y con su mera existencia me impregnan de luz cada vez que respiro.

A mi madre y a mi padre, por todo su amor y por su apoyo incondicional. Por su fuerza ante la vida.

A mis tíos porque siempre han estado pendiente de mi como unos padres más.

A mi esposa, por ser amiga, compañera, consejera, quererme, mimarme, ser especial, amarme y estar conmigo en los momentos más difíciles aun estando tan lejos pero siempre cerca.

A todos mis amigos, por ser seres humanos maravillosos.

A la hermosa obra iniciada el 1ro de enero de 1959.

También quisiera expresar mi reconocimiento y gratitud a mi tutor Yoan, por su ayuda personal y profesional en la creación de este trabajo, por el tiempo dedicado, la experiencia y los consejos.

A todos aquellos que me ayudaron a escalar la cima de este sueño, muchas gracias.

DEDICATORIA

Dedicatoria.

A mis abuelos: mis ángeles de la guarda. Que la luz de su amor infinito me ilumine siempre.

A mi valiente y hermosa madre.

A mi familia en general porque son lo que más quiero en la vida.

RESUMEN

Resumen.

A medida que la humanidad avanza, el proceso de formación revoluciona sus métodos y formas de gestión. El uso de las tecnologías de la información y las comunicaciones ha favorecido en gran medida la incorporación de nuevos espacios de aprendizaje: los espacios virtuales, aulas virtuales, tutores virtuales, entre otros

En el año 2009 se crea el proyecto de innovación pedagógica: Metodología para la Evaluación del Aprendizaje Autónomo de Idiomas que, propone como solución práctica la puesta en marcha de este Sistema Tutor Inteligente (STI). Con este sistema se persigue lograr un ambiente alternativo de evaluación, que favorezca el trabajo independiente de los estudiantes atendiendo a sus dificultades y profundizar en temas que sean de su interés para fortalecer el aprendizaje autónomo de inglés.

El presente trabajo de diploma tiene como objetivo la creación de un módulo del Tutor Virtual para la Evaluación del Aprendizaje Autónomo de Inglés (VIRTEVALL), con el cual se pretende lograr la integración de los demás módulos de VIRTEVALL. Para ello se realizó un estudio entorno a los Sistemas Tutores Inteligentes, cómo muestran la información a los estudiantes, sus características, funcionalidades, y a partir de este se determinaron las herramientas y técnicas que permiten dar solución al problema de la investigación.

PALABRAS CLAVES

Implementación, Evaluación del aprendizaje, Perfil de Usuario, Sistema Tutor Inteligente.

ÍNDICE DE CONTENIDOS.

Índice de contenidos.

Declaración de autoría	III
Agradecimientos.	IV
Dedicatoria.....	V
Resumen.	VI
Introducción.	1
1. Introducción al capítulo.....	6
1.2 Estudio de los STI.....	6
1.3 Inteligencia Artificial en los Sistemas Tutores Inteligentes.	10
1.4 Descripción de los distintos módulos de VIRTEVALL.	12
1.5 Descripción de las tecnologías a utilizar.....	13
1.5.1 Lenguaje de programación web.....	13
1.5.2 Entorno de desarrollo integrado.....	15
1.5.3 Gestor de Base de Datos.....	16
1.5.4 Marcos de Trabajo.....	18
1.6 Conclusiones del capítulo.....	20
2. Introducción al capítulo.....	21
2.1 Valoración del diseño.....	21
2.2 Técnica de inteligencia artificial implementada.....	21
2.3 Descripción de los algoritmos no triviales a implementar.	24
2.3.1 Algoritmo que le muestra todos sus datos al usuario.....	24
2.3.2 Algoritmo que muestra el andar del estudiante por el sistema.	26
2.3.3 Algoritmo que busca los objetivos de un nivel.....	27

ÍNDICE DE CONTENIDOS.

2.3.4	Algoritmo que muestra los objetivos no vencidos del nivel.	27
2.3.5	Algoritmo en el cual se le realiza el contrato al estudiante.	28
2.3.6	Algoritmo que calcula el nivel de idioma.	30
2.4	Patrones de diseño utilizados.	32
2.4.1	Patrones GRASP.	32
2.4.2	Patrones GoF.	32
2.5	Diagramas necesarios para la implementación.	33
2.5.1	Diagrama de componentes.	33
2.5.2	Diagrama de clases del diseño.	34
2.5.3	Diagrama de despliegue.	35
2.6	Estándar de codificación.	36
2.6.1	Identificadores.	37
2.6.2	Indentación.	38
2.6.3	Líneas y espacios en blanco.	39
2.6.4	Comentarios.	40
2.7	Integración mediante el perfil de usuario.	41
2.8	Conclusiones.	44
	Capítulo 3: Validación de la solución propuesta.	45
3.	Introducción.	45
3.1	Pruebas de software.	45
3.2	Automatización de pruebas.	46
3.3	Herramientas para la realización de las pruebas.	46
3.4	Pruebas de Caja Negra.	47

ÍNDICE DE CONTENIDOS.

3.4.1	Descripción del caso de prueba proyección personal.	48
3.4.2	Descripción del caso de prueba Contrato.	49
3.5	Pruebas de Caja Blanca.	51
3.5.1	Diseño de casos de prueba de Caja Blanca aplicados.	52
3.5.2	Casos de pruebas.	54
3.5.2.1	Caso de prueba para el camino básico número uno.	54
3.5.2.2	Caso de prueba para el camino básico número dos.	55
3.6	Resumen de las Pruebas.	56
3.7	Conclusiones.	57
	Conclusiones.	58
	Recomendaciones.	59
	Bibliografía	60

INTRODUCCIÓN.

Introducción.

Actualmente Cuba se encuentra inmersa en un proceso de informatización y desarrollo en todos los niveles de enseñanza. Este demanda que cada profesional que participe en él, sea una persona capaz con una preparación integral, incitándolo a llevar a cabo un proceso de superación, con el objetivo de que obtenga una mejor formación y así poder desarrollar habilidades y ampliar su marco de conocimiento.

Se puede afirmar que el dominio de las lenguas extranjeras es uno de los principales requisitos en este proceso de formación, debido a que su estudio está determinado por necesidades económicas, políticas y sociales. Estas a su vez están condicionadas por las posibilidades objetivas y subjetivas existentes en Cuba, entre las que se encuentran el poder realizar intercambios culturales y profesionales con instituciones internacionales, que a su vez son de gran ayuda. A través de estos se fomenta el desarrollo, tanto económico como intelectual, mediante los lazos de amistad y cooperación entre los diferentes sectores.

La Universidad de las Ciencias Informáticas (UCI) es una institución que no está exenta de todo este proceso, por el papel que juega en la informatización del país. Es por ello que tiene un papel protagónico en la incorporación de las Tecnologías de la Información y las Comunicaciones (TICs) en la formación de sus estudiantes. La enseñanza de las lenguas extranjeras siempre ha ocupado un lugar de vanguardia en este sentido. Por estas razones la UCI tiene como una de las tareas más urgentes la incorporación armónica de las TICs, no sólo en el proceso de enseñanza-aprendizaje, sino en la evaluación del aprendizaje autónomo.

En la Universidad de las Ciencias Informáticas, uno de los primeros pasos para hacer un mejor uso de las TICs en el proceso de evaluación del aprendizaje autónomo de idiomas, fue la creación del proyecto de innovación pedagógica Centro Virtual de Auto-aprendizaje de Lenguas Extranjeras (CEVALE). Este proyecto surgió a partir del re-diseño de la estrategia curricular de la disciplina de Idioma Inglés. Además, la UCI cuenta con un Centro de Auto-aprendizaje Virtual de Idiomas Extranjeros (EVA Moodle), en el cual estudiantes y docentes

INTRODUCCIÓN.

acuerdan, planifican, desarrollan e interactúan con el objetivo de promover el aprendizaje de los estudiantes. Sin embargo este centro no logra integrar los procesos de diagnóstico, entrenamiento y acreditación de los estudiantes debido a que es una plataforma modular y los módulos no interactúan entre ellos para exponer a los estudiantes cuáles son sus deficiencias o nivel alcanzado en la disciplina.

Para intensificar el uso de las TICs y hacer un mejor empleo de las mismas en el proceso de evaluación del aprendizaje autónomo, se plantea el uso de los Sistemas Tutores 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.

En el mundo está muy difundido el uso de los STI. En Cuba sólo se conoce el uso de dos de estos sistemas, los cuales son el Sistema Tutorial Inteligente para Diagnóstico y Tratamiento de Infecciones de Transmisión Sexual (STIITS) y el Sistema Tutorial Inteligente Multimedia Generalizado en las disciplinas de física, química, matemática y biología, desarrollado por el Instituto Superior Politécnico "José Antonio Echeverría", demostrándose así la poca difusión y uso de este tipo de medios educativos en el país.

En este contexto se comienza en el año 2009 el proyecto de innovación educativa: Metodología para la Evaluación del Aprendizaje Autónomo de Idiomas Extranjeros (VIRTEVALL). Con la puesta en marcha de este proyecto se propone lograr un ambiente alternativo de evaluación para el aprendizaje autónomo de idiomas extranjeros, que favorezca el trabajo independiente de los estudiantes atendiendo a sus dificultades y profundizar en temas que sean de su interés. Este proyecto está estructurado sobre la base del sustento teórico provisto por una tesis doctoral sobre una metodología para la evaluación del aprendizaje autónomo de idiomas extranjeros. Como aporte práctico, ofrece el STI para implementar la metodología elaborada, sustentado en el desarrollo de dos tesis de maestría que abordan la integración de las tecnologías en la gestión de la evaluación para aprendizaje autónomo de Idiomas Extranjeros, y la integración de algoritmos de inteligencia artificial en la gestión de la evaluación para aprendizaje autónomo de idiomas extranjeros.

INTRODUCCIÓN.

En el proyecto de innovación educativa VIRTEVALL se han desarrollado varios módulos de un Sistema Tutor Inteligente para el diagnóstico, el entrenamiento, la certificación y la generación de reportes de la evaluación del aprendizaje autónomo de idioma inglés, sin embargo estos módulos de manera independiente:

- ✓ Brindan información limitada y dispersa sobre el proceso de evaluación del aprendizaje autónomo de los estudiantes.
- ✓ No contemplan las características individuales de los estudiantes para la evaluación del aprendizaje autónomo.
- ✓ Limita las acciones de seguimiento y control de la evaluación por parte del estudiante y del profesor.

A partir de las limitaciones antes planteadas surge la necesidad de la creación de un módulo “Perfil de Usuario”, del cual se levantan los requisitos y se obtienen las especificaciones del mismo. Teniendo en cuenta lo anteriormente planteado el **problema a resolver** queda formulado por la siguiente interrogante: ¿Cómo contribuir a la integración de los módulos Diagnóstico, Entrenamiento, Certificación y Generador de Reportes del Sistema Tutor Inteligente?

Definiendo como **objeto de estudio** los Procesos de Desarrollo de software para la gestión de la evaluación del aprendizaje autónomo de idiomas.

Presentando como **campo de acción**, Perfil de Usuario de los Sistemas Tutores Inteligentes.

Determinando la siguiente **idea a defender**, la implementación del módulo “Perfil de Usuario” favorecerá la integración de los módulos del Tutor Virtual para la Evaluación del Aprendizaje Autónomo de Idiomas.

Trazándose el siguiente **objetivo general**, Desarrollar el módulo de Perfil de Usuario del Tutor Virtual para la Evaluación del Aprendizaje Autónomo de Idiomas.

INTRODUCCIÓN.

Para lograr lo planteado en el objetivo general se han definido un conjunto de **tareas de la investigación** que se llevarán a cabo durante el desarrollo del presente trabajo:

- ✓ Revisar antecedentes de perfiles de usuarios de otros tutores inteligentes.
- ✓ Identificar componentes existentes que agilicen el desarrollo de software educativo utilizando la tecnología web.
- ✓ Utilizar reglas de Inteligencia Artificial para contribuir a la integración de los módulos.
- ✓ Codificar el módulo “Perfil de Usuario”.
- ✓ Desarrollar casos de Prueba que certifiquen las reglas de Inteligencia Artificial y el código del módulo Perfil de Usuario.

Métodos científicos de la investigación.

Para el desarrollo de la investigación se utilizaron métodos teóricos y empíricos.

Métodos teóricos:

- ✓ **Análisis histórico-lógico:** Este método permitirá realizar un estudio entorno a los STI, cómo muestran la información a los estudiantes, sus características y sus funcionalidades.
- ✓ **Analítico - Sintético:** Este método permitirá después de realizar un estudio entorno a los STI, determinar los principales métodos y algoritmos que son usados a nivel mundial en estos sistemas y que pueden ser de utilidad en la implementación del módulo.

Métodos empíricos:

- ✓ **Entrevista:** Se realizaron múltiples entrevistas no formales a los clientes, líderes de proyecto, analistas en aras de obtener información referente al funcionamiento del sistema.

Posibles resultados:

INTRODUCCIÓN.

Un módulo implementado con tecnología web, el cual favorecerá la evaluación del aprendizaje autónomo de inglés.

Estructuración del contenido.

El presente trabajo está conformado por 3 capítulos que se describen a continuación.

Capítulo 1. Fundamentación teórica: Se realiza un estudio entorno a los STI, cómo muestran la información a los estudiantes, sus características y sus funcionalidades. Se hace un estudio del uso de las técnicas de Inteligencia Artificial empleadas en los STI. Se describe brevemente cada módulo del STI VIRTEVALL, además se describen las herramientas a utilizar para solucionar el problema planteado.

Capítulo 2. Descripción y análisis de la solución propuesta: Se describen algunos algoritmos a implementar y se analiza la complejidad de los mismos, así como las clases que modelan la solución y las principales funcionalidades de cada una de ellas. Y como se evidencia la integración a través del módulo Perfil de Usuario.

Capítulo 3. Validación de la solución propuesta: Aborda el diseño de las pruebas de unidad que permitan validar la solución propuesta, detallando de las mismas, su objetivo, alcance y tipo; al igual que los valores utilizados para la ejecución de las pruebas que tributan a obtener un sistema con la calidad requerida.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1. Introducción al capítulo.

En este capítulo se hace un estudio de los Sistemas Tutores Inteligentes (STI), de las técnicas de Inteligencia Artificial (IA) que se implementan, una descripción breve de las funcionalidades de cada módulo del Tutor Virtual para la Evaluación del Aprendizaje Autónomo de Idiomas (VIRTEVALL). Además, se describen las tecnologías que en términos de desarrollo de aplicaciones web se emplearon: lenguaje de programación, Entorno de Desarrollo Integrado (IDE), Gestor de Base de Datos (GBD), marcos de trabajo; considerando sus ventajas para la integración de los distintos módulos del STI.

Se asumen las concepciones de las tesis de Arquitectura y Análisis y Diseño del STI VIRTEVALL que anteceden a la actual investigación.

1.2 Estudio de los STI.

Los Sistemas Tutores Inteligentes comenzaron a desarrollarse en los años ochenta con la idea de poder impartir el conocimiento usando alguna forma de inteligencia para poder asistir y guiar al estudiante en su proceso de aprendizaje. Se buscó emular el comportamiento de un tutor humano a través de un sistema que pudiera adaptarse al comportamiento del estudiante, identificando la forma en que el mismo resuelve un problema a fin de poder brindarle ayudas cognitivas cuando lo requiera. (Sierra. E y otros, 2004)

Algunas definiciones de tutor inteligente que revelan su esencia son las siguientes:

“Sistemas que modelan la enseñanza, el aprendizaje, la comunicación y el dominio del conocimiento del especialista y el entendimiento del estudiante sobre ese dominio”. (Wolf. B, 1984)

“Es un sistema de software que utiliza técnicas de inteligencia artificial (IA) para representar el conocimiento e interactúa con los estudiantes para enseñárselo”. (Vanlehn. K, 1988)

“Un sistema que incorpora técnicas de IA (Inteligencia Artificial) a fin de crear un ambiente que considere los diversos estilos cognitivos de los alumnos que utilizan el programa”. (Giraffa. L y otros, 1997)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

En todas las definiciones se aprecia que coinciden en el uso de técnicas de IA, así como la intención de simular a un tutor humano en la interacción con los estudiantes. Las últimas tendencias hacen referencia a la utilización de los estilos cognitivos o de aprendizaje para personalizar la tutoría.

Según Cataldi (2009) los STI permiten la emulación de un tutor humano para determinar qué enseñar, cómo enseñar y a quién enseñar a través de:

- ✓ **Módulo del Dominio:** define el dominio del conocimiento (ver Figura 1).
- ✓ **Módulo del Estudiante:** es capaz de definir el conocimiento del estudiante en cada punto durante la sesión de trabajo.
- ✓ **Módulo del Tutor:** genera las interacciones de aprendizaje basadas en las discrepancias entre el especialista y el estudiante.
- ✓ **Módulo Interfaz de Usuario:** permite la interacción del estudiante con un STI de una manera eficiente.

A través de la interacción entre los módulos básicos, los STI son capaces de determinar lo que sabe el estudiante y cómo va en su progreso, por lo que la enseñanza se puede ajustar según las necesidades del estudiante, sin la presencia de un tutor humano.

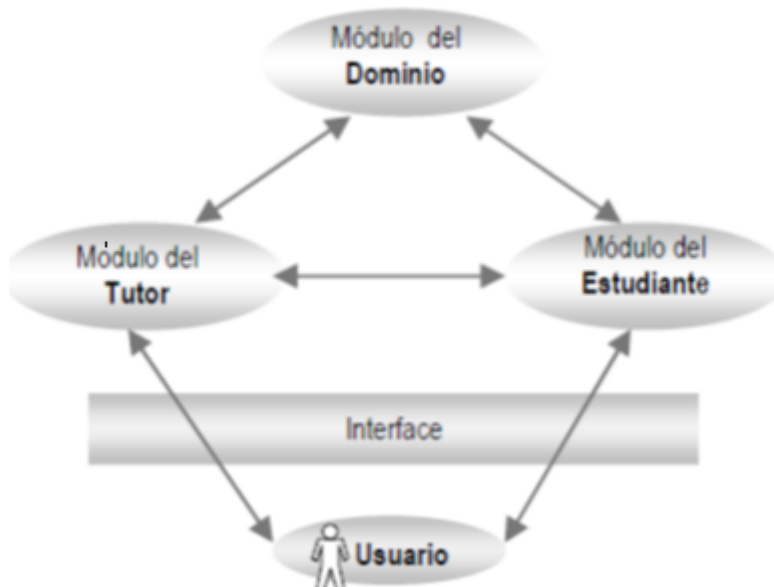


Figura 1: Interacción de los módulos de un Sistema Tutor Inteligente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

El **Módulo Tutor** es quien define y aplica una estrategia pedagógica de enseñanza, contiene los objetivos a ser alcanzados y los planes utilizados para alcanzarlos. Selecciona los problemas, monitorea el desempeño, provee asistencia y selecciona el material de aprendizaje para el estudiante, integra el conocimiento acerca del método de enseñanza, las técnicas didácticas y del dominio a ser enseñado (ver Figura 2). (Cataldi. Z y otros, 2009)

El mismo consta de:

- a) *Protocolos Pedagógicos*: estos son almacenados en una Base de Datos, con un gestor para la misma.
- b) *Planificador de Lección*: se encarga de organizar los contenidos de la misma.
- c) *Analizador de Perfil*: analiza las características del alumno, seleccionando la estrategia pedagógica más conveniente.

El **Módulo Estudiante** tiene por objetivo realizar el diagnóstico cognitivo del alumno, y el modelado del mismo para una adecuada retroalimentación del sistema. Para el módulo estudiante se han planteado los siguientes sub-módulos. (Ver Figura 2): (Cataldi. Z y otros, 2009)

- a) *Estilos de aprendizaje*: está compuesto por una Base de Datos con los estilos de aprendizajes disponibles en el sistema, los métodos de selección de estilos y las características de cada uno de ellos. 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.
- b) *Estado de conocimientos*: contiene el mapa de conocimientos obtenido inicialmente a partir del módulo del dominio y que el actualizador de conocimientos irá modificando progresivamente a través de los resultados obtenidos en las evaluaciones efectuadas por el módulo del tutor quien le enviará dichos resultados procesados.
- c) *Perfil psico-sociológico del estudiante*: para determinar el perfil psico-sociológico se usa la Teoría de las Inteligencias Múltiples de Gardner

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

(1993, 2001) quien señala “*no existe una inteligencia única en el ser humano, sino una diversidad de inteligencias que evidencian las potencialidades y aspectos más significativos de cada individuo, en función de sus fortalezas y debilidades para la expansión de la inteligencia*”. Señala que las inteligencias trabajan juntas para:

1. Resolver problemas cotidianos.
2. Crear productos.
3. Ofrecer servicios dentro del propio ámbito cultural.

El **Módulo Dominio** tiene el objetivo global de almacenar todos los conocimientos dependientes e independientes del campo de aplicación del STI (ver Figura 2). (Cataldi. Z y otros, 2009)

Como los demás módulos, este también posee sub-módulos que le ayudan en su funcionamiento, ellos son:

- a) *Parámetros Básicos del Sistema*: se almacenan en una Base de Datos con un gestor para la misma.
- b) *Conocimientos*: son los contenidos que deben cargarse en el sistema, a través de los conceptos, las preguntas, los ejercicios, los problemas y las relaciones.
- c) *Elementos Didácticos (Contextos)*: son las imágenes, videos y sonidos. Material multimedia que se requiere para brindar al estudiante varios canales de percepción de la información.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

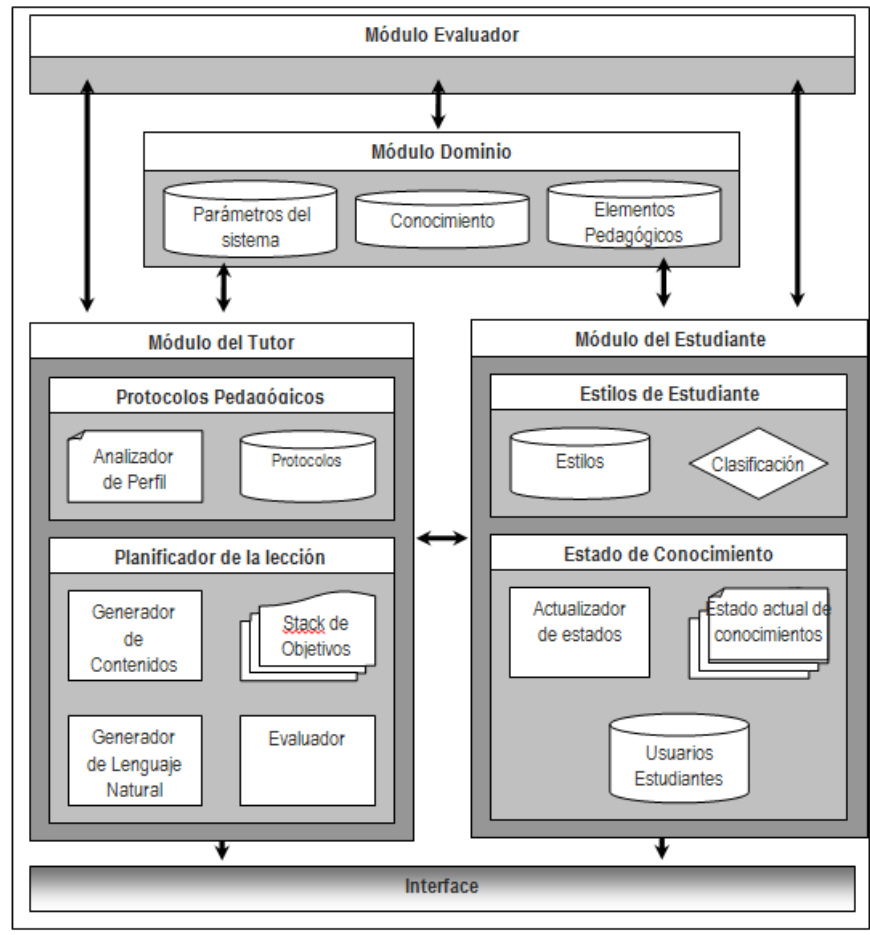


Figura 2: Esquema de un STI con sus módulos principales.

Además de los tres módulos estándares, presentados en la Figura 2 y descritos anteriormente, existe un **Módulo Interfaz de Usuario** que tiene como objetivos: generar las salidas correctas para el estudiante, interpretar sus respuestas, organizarlas y pasarlas al sistema de tutoría. Este módulo es el que ha recibido menos atención por los problemas de comprensión de lenguaje natural. Es a través de este módulo que se brinda toda evidencia del desarrollo de las habilidades de los estudiantes generada por los restantes módulos del STI, **Módulo Dominio**, **Módulo Tutor**, **Módulo Estudiante**, de la cual el estudiante se nutre para determinar el nivel de idioma que posee y cómo va en su progreso.

1.3 Inteligencia Artificial en los Sistemas Tutores Inteligentes.

En el contexto de los sistemas inteligentes se encuentra la utilización de técnicas de IA para determinar futuros rendimientos de los estudiantes, obtener

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

el nivel de conocimiento asociado al estudiante, y para determinar patrones de aprendizaje en los estudiantes. Entre las técnicas de inteligencia artificial más explotadas en los Sistemas Tutores Inteligentes encontramos las redes neuronales, las redes bayesianas y la minería de datos. (Cataldi. Z y otros, 2009)

✓ **Redes Neuronales.**

Las redes neuronales son *“interconexiones masivas en paralelo de elementos simples y que responden a una cierta jerarquía intentando interactuar con los objetos reales tal como lo haría un sistema neuronal psicológico”*(Kohonen. T, 1998). Las redes neuronales poseen la característica de asimilar conocimiento en base a las experiencias mediante la generalización de casos (Haykin. S, 1999).

Por ejemplo, para efectuar la predicción del rendimiento académico, se puede usar una red neuronal de tipo backpropagation tomando como datos de entrada los resultados de las evaluaciones parciales desagregados en dos formas.

- a) Tomando el caso de resolución por ejercicios.
- b) Tomando ejercicios en función de los logros cognitivos.

Usando los datos provenientes de las evaluaciones parciales de los estudiantes se pueden predecir futuros rendimientos.

✓ **Redes Bayesianas.**

Las redes bayesianas son herramientas estadísticas orientadas a la inferencia probabilística y en el ámbito de la tutorización electrónica se pueden utilizar para modelar la incertidumbre asociada al estudiante y su nivel de conocimientos. Los algoritmos genéticos, se fundamentan en el concepto biológico de la evolución natural y son utilizados en procesos de optimización (Falkenauer. E, 1999). Se fundamentan en los mecanismos de la selección natural, por los que sólo sobreviven los individuos más aptos, luego de la interacción entre los mismos, pertenecientes a una población de posibles soluciones.

✓ **Minería de Datos.**

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

La minería de datos se centra en la búsqueda de patrones sugerentes y regularidades importantes en grandes Bases de Datos, denominado conocimiento cualitativo. La minería se puede aplicar con métodos de sistemas inteligentes y otros métodos asociados, para descubrir y detallar patrones presentes en los datos. Se pueden obtener agrupaciones en un conjunto de datos, sin tener relaciones o clases predefinidas, basándose en la similitud de los valores de los atributos de los distintos datos. La minería de datos se puede aplicar incluyendo algoritmos de inducción, algoritmos genéticos, redes neuronales y redes bayesianas; de acuerdo al problema a resolver.

1.4 Descripción de los distintos módulos de VIRTEVALL.

- ✓ **Módulo Diagnóstico:** Este sub-sistema es el encargado de la gestión de los diagnósticos que el sistema brinda para que un estudiante realice. Este está compuesto por dos sub-módulos: diagnóstico de las características de la personalidad y posicionamiento de nivel de idioma.
- ✓ **Módulo Entrenamiento:** Es a través de este módulo que el estudiante ya autenticado en el sistema realiza las actividades propuestas por el mismo para desarrollar el entrenamiento en la evaluación del aprendizaje autónomo en el nivel de idioma que arrojó el diagnóstico de posicionamiento.
Este sub-sistema está compuesto por el sub-módulo Actividades que es donde se gestionan todas las actividades que el estudiante realiza.
- ✓ **Módulo Certificación:** Este pequeño sub-sistema es el encargado de la gestión y realización de los exámenes por parte de los estudiantes para la acreditación de un nivel de idioma.
- ✓ **Módulo Generador de Reportes:** Módulo encargado de generar reportes de cada estudiante o de datos específicos en el sistema, dígame nombre de todos los estudiantes, notas de todos o de un estudiante en específico, dirección, usuario, inclusive todos los datos de una o varias tablas de la Base de Datos del sistema. Estos reportes son generados por un profesor para nutrirse de la información solicitada y poder actuar en dependencia de los resultados obtenidos en la misma y así poder, por ejemplo, darle seguimiento a los estudiantes y ver sus deficiencias y fortalezas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

- ✓ **Módulo Perfil de Usuario:** El Perfil de Usuario es el que interactúa directamente con el usuario registrado en el sistema. Este sub-sistema se encarga de guiar al estudiante para interactuar con el sistema, es el que le propone las actividades más acorde al nivel de idioma que posee el usuario. Se encarga de la integración de los demás módulos del sistema y brinda información del desarrollo de las habilidades de los estudiante, generada por los restantes módulos del Sistema Tutor Inteligente.

1.5 Descripción de las tecnologías a utilizar.

Las tecnologías empleadas durante la presente investigación fueron previamente seleccionadas por el grupo de arquitectos del proyecto VIRTEVALL, quienes asumieron para su selección algunos criterios, entre los que se destacan: el uso de buenas prácticas de diseño web, que las herramientas se encuentren bajo licencia de Software Libre, que sean multiplataforma, flexibles ante cambios y que exista una integración entre cada una de ellas. (Rodríguez. O, 2010)

1.5.1 Lenguaje de programación web.

El lenguaje de programación seleccionado por parte del arquitecto para la realización e implantación del STI VIRTEVALL fue PHP en su versión 5. PHP es un lenguaje interpretado de alto nivel impregnado en páginas HTML y ejecutado en el servidor. El 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. PHP5 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.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

- ✓ **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

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

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.

1.5.2 Entorno de desarrollo integrado.

Un Entorno de Desarrollo Integrado (en inglés Integrated Development Environment) 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)

NetBeans fue el IDE seleccionado para desarrollar el STI en su versión 6.8. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java. NetBeans IDE es un producto libre y gratuito sin restricciones de uso y es multiplataforma. (Rodríguez. O, 2010)

Características de NetBeans. (Rodríguez. O, 2010)

- ✓ **Creación de Proyectos PHP:** NetBeans nos provee de una estructura para los proyectos que podemos crear junto a este IDE, nos propone una estructura para organizar nuestro código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS.
- ✓ **Integración con Symfony y ZenFramework:** Su integración con estos populares Framework de PHP, es una de las características que permite utilizar a NetBeans. Realizar aplicaciones con estos Framework es muy ágil, mucho más si se integran con una herramienta que lo agiliza.
- ✓ **Editor de Código Fuente:** A lo largo de todo este tiempo, se notó la mejora en su editor, sobre todo en el editor de PHP, es mucho más ágil y a

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

la vez robusto, contiene más ayuda en línea, reconocimiento de sintaxis y todo lo que provee la última versión de PHP, la 5.3.

- ✓ **Integración con PHP Unit Testing:** Es posible crear test con PHPUnit, para diferentes funciones, luego realizar la comprobación y ver todos los resultados. En las propiedades PHPUnit puede definir una configuración personalizada de archivos XML, un archivo de arranque para las opciones de línea de comandos, o una serie de pruebas a medida, o puede que el IDE genere el código para usted.
- ✓ **Depuración de PHP:** NetBeans integra muy bien la utilización Xdebug, gracias a esto podemos inspeccionar y examinar cada variable local, establecer puntos de interrupción y evaluar el código en nuestra lógica. El IDE de NetBeans para PHP también ofrece la línea de comandos de depuración: La salida del programa PHP aparece en una pantalla de línea de comandos en el IDE de sí mismo y se puede inspeccionar el código HTML generado sin tener que cambiar a un navegador.
- ✓ **Integración con Sistemas de Control de Versiones:** Esta es una de las condiciones necesarias para los proyectos y es la posibilidad de contar con la integración de sistemas de control de versiones.

1.5.3 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. (PostgreSQL)

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.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Se determinó que el Gestor de Base de Datos (GBD) para almacenar toda la información generada por el STI fuera PostgreSQL en su versión 8.4.3. (Rodríguez. O, 2010)

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 Projects 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 generará 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.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.5.4 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)

¿Qué ventajas tiene utilizar un marco de trabajo?

Las que se derivan de utilizar un estándar entre otras:

- ✓ El programador no necesita plantearse una estructura global de la aplicación, sino que el marco de trabajo le proporciona una estructura que hay que "rellenar".
- ✓ Facilita la colaboración. Cualquiera que haya tenido que "pelearse" con el código fuente de otro programador (¡o incluso con el propio, pasado algún tiempo!) sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos.
- ✓ Es más fácil encontrar herramientas (utilidades, bibliotecas) adaptadas al marco de trabajo concreto para facilitar el desarrollo.

Se seleccionaron dos marcos de trabajo para el desarrollo del STI: **Symfony** para la parte del desarrollo de los módulos **Dominio**, **Tutor**, **Estudiante** y **Sencha Ext JS** para el desarrollo de la **Interfaz de Usuario**.

• **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.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

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.
- **Sencha ExtJs** (Ecured, 2011)

ExtJS es una biblioteca Javascript que permite construir aplicaciones complejas en internet además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales.

Esta biblioteca incluye:

- ✓ Componentes UI del alto performance y personalizables.
- ✓ Modelo de componentes extensibles.
- ✓ Un API fácil de usar.
- ✓ Licencias Open Source (GPL) y comerciales.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

ExtJS ha marcado la diferencia en la preferencia de muchos desarrolladores. Y no solo para aplicaciones web sino que ya es útil para productos de alta demanda en el mercado como iPhone e IPAD.

- ✓ Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos.
- ✓ Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores (Firefox, IE, Safari, Opera etc.).
- ✓ El funcionamiento de las ventanas flotantes lo pone por encima de cualquier otro.
- ✓ Relación entre Cliente-Servidor balanceado: Se distribuye la carga de procesamiento entre, permitiendo que el servidor pueda atender más clientes al mismo tiempo.
- ✓ Eficiencia de la red: Disminuye el tráfico en la red pues las aplicaciones cuentan con la posibilidad de elegir qué datos desea transmitir al servidor y viceversa (Criterio este que puede variar con el uso de aplicaciones de pre-carga).
- ✓ Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.

1.6 Conclusiones del capítulo.

En este capítulo se profundiza y caracteriza a los STI y cómo estos muestran la información a los estudiantes pasando a través de sus diferentes módulos. Se plasman las técnicas de IA utilizadas en los STI y para qué se utilizan. Se hace una breve descripción de cada módulo del STI, sus características y sus deficiencias así como de las tecnologías a emplear en el desarrollo del STI haciendo énfasis en las características y ventajas de cada uno de estos y teniendo en cuenta las concepciones de las tesis que anteceden a la actual investigación.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

2. Introducción al capítulo.

En este capítulo se realiza un análisis crítico y valorativo del diseño propuesto por la tesis de Análisis y Diseño que antecede a esta investigación para así comenzar el proceso de implementación. Se hace un análisis de la complejidad y funcionamiento de los algoritmos más importantes para la implementación del sistema, así como la técnica de IA empleada para manejar los datos en el sub-sistema. Se presenta la descripción de las clases fundamentales que definen el comportamiento del sistema con sus atributos y métodos esenciales. Finalmente se definen los estándares de codificación a utilizar en la implementación del sistema.

2.1 Valoración del diseño.

Del diseño propuesto en la tesis de Análisis y Diseño se pueden identificar las funcionalidades a implementar para que el sistema funcione correctamente, basándose en la descripción detallada de los casos de uso, se puede establecer una estrategia de trabajo que permita la implementación de la aplicación. Tomando como base la propuesta de las herramientas a usar, descritas en el capítulo anterior.

2.2 Técnica de inteligencia artificial implementada.

La técnica de inteligencia artificial implementada en el módulo Perfil de Usuario es la red bayesiana. En la red se tienen en cuenta cuatro variables que son las que rigen el proceso de evaluación del aprendizaje. Partiendo de que se desean obtener las evidencias que reflejan el nivel del estudiante en la evaluación del idioma inglés, las variables seleccionadas fueron: el nivel, la fase, la habilidad y los objetivos. Se tienen en cuenta las habilidades básicas del idioma, como son: comprensión auditiva, comprensión lectora, expresión escrita y expresión oral. Los objetivos son definidos para cada habilidad en una fase y un nivel de idioma. (Mayet. M, 2010)

Viendo cada variable por separado, se tendrá que cada objetivo estará representado por la letra O, la misma tendrá un valor binario:

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

$$O = \begin{cases} 0, & \text{si el objetivo no fue vencido} \\ 1, & \text{si el objetivo fue vencido} \end{cases}$$

La variable habilidad se representará con la letra H y estará formada por un par (O, W), donde:

- ✓ O es un conjunto de objetivos que serán independientes entre sí.
- ✓ W es el vector de pesos que representa la importancia de cada objetivo en la habilidad a la que pertenece. De lo anterior se tendrá:

$$\sum_{i=1}^n W_i = 1$$

Teniendo en cuenta lo antes expuesto una habilidad estará dada de la siguiente forma:

$$H = \sum_{i=1}^n W_i O_i$$

La fase se representará con la letra F y estará formada por un par (H, Z), donde:

- ✓ H son un conjunto de habilidades necesarias para vencer una fase, que serán independientes entre sí.
- ✓ Z es el vector de pesos que representa la importancia de cada habilidad en el cumplimiento de la fase. De lo anterior se tendrá:

$$\sum_{i=1}^n Z_i = 1$$

Teniendo en cuenta lo antes expuesto una fase estará dada de la siguiente forma:

$$F = \sum_{i=1}^n H_i Z_i$$

El nivel se nombrará con la letra N y está constituido por un par (F, X) donde:

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

- ✓ F son un conjunto de fases necesarias para vencer un nivel, que serán independientes entre sí.
- ✓ X es el vector de pesos que representa la importancia de cada fase en el cumplimiento del nivel. De lo anterior se tendrá:

$$\sum_{i=1}^n X_i = 1$$

Para la representación del conocimiento en un nivel se hará de la siguiente forma:

$$N = \sum_{i=1}^n F_i X_i$$

A continuación se muestra como queda el esquema general del perfil del estudiante. (Solano, 2010)



Figura 3: Esquema general de la red bayesiana para el perfil del estudiante.

La codificación de esta técnica de inteligencia artificial se contempla en la función EstadoProceso de la clase PerfilAction del módulo Perfil de Usuario.

A continuación se pone un ejemplo de como es que se le muestra al usuario el conocimiento adquirido durante cada momento que interactúa con el sistema.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

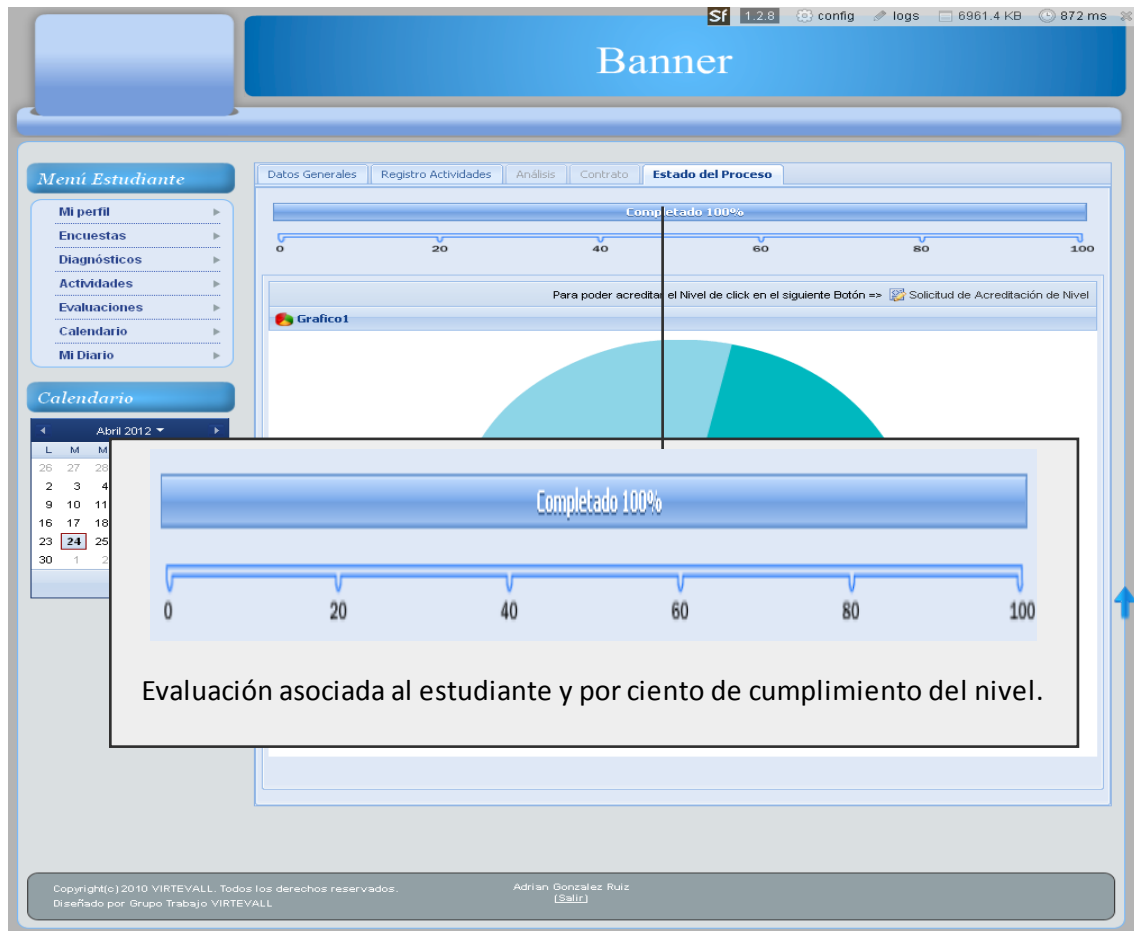


Figura 4: Conocimiento adquirido por el estudiante.

2.3 Descripción de los algoritmos no triviales a implementar.

En esta sección se hará un análisis de aquellos algoritmos que resultan importantes para la implementación del módulo o que su codificación es compleja. Se mostrará el código fuente de los mismos y se describirán los pasos principales que componen el algoritmo.

2.3.1 Algoritmo que le muestra todos sus datos al usuario.

Este algoritmo está contenido en la función Perfil de la clase PerfilActions del módulo Perfil de Usuario. Su objetivo consiste en mostrarle todos los datos al usuario que hasta el momento el sistema a asimilado sobre él, datos que se irán modificando y adicionando a medida que el estudiante valla avanzando e interactuando con el sistema.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

```
public function executePerfil() {
    $idUserario = $this->getUser()->getAttribute('id_usuario_activo');
    $datosUsuario = TusuarioPeer::toJsonPerfil($idUserario);
    $this->renderText($datosUsuario);
    return sfView::NONE;
}

public static function toJsonPerfil($idUserario) {
    $usuario = TusuarioPeer::obtenerUsuarioID($idUserario);
    $nombre = $usuario->getNombre(). ' '. $usuario->getApellido1(). ' '.
    $usuario->getApellido2();
    $datos = null;
    $ocupacion = null;

    if(TprofesorPeer::obtener($idUserario) != null) {
        $ocupacion = 'Profesor';
        $usuario = TprofesorPeer::obtener($idUserario);
        $datos = $usuario->getAsignaturaImparte();
    }
    else if(TalumnoPeer::obtener($id_usuario) != null) {
        $ocupacion = 'Alumno';
        $usuario = TalumnoPeer::obtener($idUserario);
        $datos = $usuario->getFacultad(). '#'. $usuario->getGrupo(). '#'.
        $usuario->getAnno();
    }
    else if(TtrabajadorPeer::obtener($idUserario) != null) {
        $ocupacion = 'Trabajador';
        $usuario = TtrabajadorPeer::obtener($idUserario);
        $datos = $usuario->getArea();
    }
    $usuario = TestudiantePeer::obtenerAlumnoId($idUserario);
    $estilo = $usuario->getEstiloAprendizaje();
    $estrategia = $usuario->getEstrategiaAprendizaje();
    $perfil = $usuario->getPerfilInteligencia();
    $nivelAE = $usuario->getNivelAutoevaluacion();
    $nivelAC = $usuario->getNivelAcreditado();
    $nivelEN = $usuario->getNivelEntrenamiento();
}
```

Figura 5: Función Perfil de la clase PerfilActions del módulo Perfil.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

2.3.2 Algoritmo que muestra el andar del estudiante por el sistema.

Algoritmo que está contenido en la función Registros de la clase PerfilActions del módulo Perfil de Usuario. Su objetivo consiste en mostrarles al profesor y al estudiante todo lo que ha hecho este en el sistema. Es más bien una especie de traza de la que el profesor por la parte administrativa se puede servir para ver qué está haciendo el estudiante en la aplicación, y al estudiante le sirve para ver cuánto ha hecho o que le falta por hacer.

```
public function executeRegistros() {  
    $idAlumno = $this->getUser()->getAttribute('id_usuario_activo');  
    $registros = TregistroActividadesPeer::obtenerRegistros($idAlumno);  
    $this->renderText(TregistroActividadesPeer::toJson($registros));  
    return sfView::NONE;  
}  
  
public static function toJson($registros) {  
    $json = array();  
    $datosregistros = array();  
    $json["success"] = true;  
    $json["count"] = count($registros);  
  
    foreach ($registros as $index => $obj )  
    {  
        $datosregistros[] = array( "id"=> $obj->getIdRegistro(),  
                                   "fecha"=> $obj->getFechaAcceso(),  
                                   "elemento"=> $obj->getElementoAccedido(),  
                                   "inicio"=> $obj->getHoraComienzo($format = 'H:i:s'),  
                                   "fin"=> $obj->getHoraFin($format = 'H:i:s'));  
    }  
  
    $json["registros"] = $datosregistros;  
    return json_encode($json);  
}
```

Figura 6: Función Registros de la clase PerfilActionsl.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

2.3.3 Algoritmo que busca los objetivos de un nivel.

Algoritmo que está contenido en la función `ObjetivosNivel` de la clase `PerfilActions` del módulo `Perfil`. Su objetivo consiste en buscar todos los objetivos del nivel en que se encuentra un estudiante dado para su posterior visualización y conocimiento.

```
function executeObjetivosNivel() {
    $idAlumno = $this->getUser()->getAttribute('id_usuario_activo');
    $alumno = TestudiantePeer::obtenerAlumnoId($idAlumno);
    $nivel = $alumno->getNivelEntrenamiento();
    $fase = Array();

    if($nivel == 1){
        $fase[0] = ($nivel * 2)-1;
        $fase[1] = $nivel * 2;
    }
    if($nivel == 2){
        $fase[0] = ($nivel * 2)-1;
        $fase[1] = $nivel * 2;
    }
    if($nivel == 3){
        $fase[0] = ($nivel * 2)-1;
        $fase[1] = $nivel * 2;
    }

    $objetivos = TobjetivoPeer::objetivosFase($fase[0]);
    $objetivosF2 = TobjetivoPeer::objetivosFase($fase[1]);

    foreach ($objetivosF2 as $key => $value)
    {
        $objetivos[] = $value;
    }

    $this->renderText(TobjetivoPeer::toJson($objetivos));
    return sfView::NONE;
}
```

Figura 7: Función `ObjetivosNivel` de la clase `PerfilActions`.

2.3.4 Algoritmo que muestra los objetivos no vencidos del nivel.

Algoritmo que está contenido en la función `ObjetivosNCumplidos` de la clase `PerfilActions` del módulo `Perfil de Usuario`. Su objetivo consiste en buscar todos los objetivos no vencidos del estudiante en un nivel determinado para que el estudiante sepa cuáles son esos objetivos y trabajar en base a vencerlos. En

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

caso de que el estudiante no tenga ningún objetivo no vencido se le envía un mensaje de notificación.

```
function executeObjetivosNCumplidos() {
    $idAlumno = $this->getUser()->getAttribute('id_usuario_activo');
    $alumno = NestudianteObjetivoPeer::objetivosNCumplidos($idAlumno);
    for ($i = 0; $i < count($alumno); $i++){
        if(!$alumno[$i]->getVencido()){
            $objetivos[] = TobjetivoPeer::obteneridObjetivo(
                $alumno[$i]->getIdObjetivo());
        }
    }
    if (count($alumno) != 0)
        $this->renderText(TobjetivoPeer::toJson($objetivos));
    else {
        $this->renderText(' {"count":' . 0 . ', "objetivos": [
            { "num":null, "id_objetivo":' . 0 . ', "descripcion":
                "Feliciades tiene todos los Objetivos Vencidos"} ] }');
    }
    return sfView::NONE;
}
```

Figura 8: Función ObjetivosNCumplidos de la clase PerfilActions.

2.3.5 Algoritmo en el cual se le realiza el contrato al estudiante.

Algoritmo que está contenido en la función Contrato de la clase PerfilActions del módulo Perfil de Usuario. Su objetivo consiste en mostrarle al estudiante su proyección personal y el tiempo que va a dedicar para darle cumplimiento a la misma. Esta proyección es seleccionada por el estudiante a partir de los objetivos no vencidos del nivel con una prioridad puesta por él mismo a cada objetivo para seleccionar por cuál comenzar a la hora de vencerlos. Por cada objetivo no cumplido se le muestran un grupo de actividades propuestas para ese objetivo. Cada objetivo mostrado en el contrato define una fecha de inicio y una fecha de fin que no es más que el intervalo de tiempo en que el estudiante cree que le va a vencer ese objetivo resolviendo las actividades propuestas

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

para ese objetivo. Todo este conjunto forma el contrato en sí, el cual tiene también una fecha de inicio y una fecha de fin, que va a ser el intervalo de tiempo en que el estudiante va a realizar todas las actividades propuestas para cada objetivo. El contrato es aceptado por el estudiante y este comienza a realizar las actividades propuestas.

```
function executeContrato(){
    $idAlumno = $this->getUser()->getAttribute('id_usuario_activo');
    $params = $this->getRequest()->getParameter('contrato');
    $selectedData = json_decode($params);
    $fechaActual = date("Y-m-d");
    $bandera = true;
    $fechaFin = $fechaActual;

    for($j = 0; $j < count($selectedData); $j++){
        if($selectedData[$j][0] > $fechaFin)
            $fechaFin = $selectedData[$j][0];
        if($selectedData[$j][1] > $fechaFin){
            $fechaFin = $selectedData[$j][1];
        }
    }

    $i = 0;
    while($bandera && $i < count($selectedData)){
        if($selectedData[$i][0] < $fechaActual){
            $bandera = false;
            $this->renderText("
                {success: 'false', respuesta: {
                reason: " . json_encode('
                Cada Fecha debe ser mayor que la del dia actual: '.
                $fecha_actual) . "}}");
        }
        if($selectedData[$i][0] > $selectedData[$i][1]){
            $bandera = false;
            $this->renderText("{success: 'false', respuesta: {
                reason: " . json_encode('
```

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

```
Verifique Las fechas,  
las Fechas de Inicio deben ser menor que las Fechas de Fin') .  
    "}}");  
  
    }  
    $i++;  
  }  
  
  if($bandera){  
    try{  
      TcontratoPeer::InsertarContrato($fechaActual, $fechaFin , $idAlumno);  
      $idcontrato = TcontratoPeer::ObtenerIdContrato($idAlumno);  
      for($i = 0; $i < count($selectedData); $i++){  
        NcontratoObjetivoPeer::InsertarContratoObjetivo($selectedData[$i][0],  
          $selectedData[$i][1], $selectedData[$i][2], $idcontrato);  
      }  
      $this->renderText("{success: 'true', respuesta: {  
        reason: " . json_encode('Contrato Aceptado, Su Contrato es Valido desde:' .  
          $fechaActual . ' hasta ' . $fechaFin) . "}}");  
    }  
    catch (Exception $exc) {  
      $error = json_encode($exc->getMessage());  
      $this->renderText("{success: 'false', respuesta: { reason: " .  
        $error . "}}");  
    }  
  }  
  return sfView::NONE;  
}
```

Figura 9: Función Contrato de la clase PerfilActions.

2.3.6 Algoritmo que calcula el nivel de idioma.

Algoritmo que está contenido en la función EstadoProceso de la clase PerfilActions del módulo Perfil. Su objetivo consiste en calcular a través de la implantación de una red bayesiana qué por ciento de cumplimiento de cada

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

objetivo tiene un estudiante en el nivel en que se encuentra. Luego de obtener el por ciento de cumplimiento de los objetivos del nivel se decide si el estudiante está apto o no para realizar un examen de acreditación de la disciplina.

```
function executeEstadoProceso() {
    $F = Array();
    $X = Array();
    $X[0] = 0; $X[1] = 0;
    $idAlumno = $this->getUser()->getAttribute('id_usuario_activo');
    $alumno = TestudiantePeer::obtenerAlumnoId($idAlumno);
    $N = 0;
    if($alumno->getFase()%2 == 0){
        $F[] = PerfilActions::CalcularFase($alumno->getFase()-1,
            $alumno->getIdAlumno());
        $F[] = PerfilActions::CalcularFase($alumno->getFase(),
            $alumno->getIdAlumno());
        $X[0] = TfasePeer::obteneridFase(
            $alumno->getFase()-1->getPrioridad());
        $X[1] = TfasePeer::obteneridFase(
            $alumno->getFase()->getPrioridad());
    }
    else
    {
        $F[] = PerfilActions::CalcularFase($alumno->getFase(),
            $alumno->getIdAlumno());
        $X[0] = TfasePeer::obteneridFase(
            $alumno->getFase()->getPrioridad());
    }

    for($l = 0; $l < count($F);$l++){
        $N = $N + $F[$l] * $X[$l];
    }
    $this->renderText(json_encode($N));
    return sfView::NONE;
}
```

Figura 10: Función EstadoProceso de la clase PerfilActions.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

2.4 Patrones de diseño utilizados.

2.4.1 Patrones GRASP.

Experto: La implementación que realiza Symfony de la arquitectura MVC incluye varias clases, en las cuales se evidencia el uso de este patrón:

- ✓ **sfController:** Clase perteneciente al controlador frontal que se encarga de decodificar la petición y transferirla a la acción correspondiente.
- ✓ **sfRequest:** Almacena los elementos que forman la petición (parámetros, cookies, cabeceras, etc.)
- ✓ **sfResponse:** Contiene las cabeceras de la respuesta y los contenidos. El contenido de este objeto se transforma en la respuesta HTML que se envía al usuario.

Alta Cohesión: En cada clase Actions se definen las acciones para las plantillas, además estas colaboran con otras para realizar diferentes operaciones, se instancian objetos, y se acceden a las propiedades. En una Actions se utilizan diferentes funcionalidades estrechamente relacionadas entre sí, lo que proporciona un software flexible ante los cambios.

Controlador: Todas las peticiones Web son manejadas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

2.4.2 Patrones GoF.

En la categoría Creacionales:

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En las acciones se usan los métodos **getRequest()**, **getUser()**, esto se debe a que, en la acción, el método **getContext()**, guarda una referencia a todos los objetos del núcleo del framework Symfony, estos métodos pueden ser accedidos desde la vista y desde el controlador.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

En la categoría Estructurales:

Decorator (Envoltorio): Añade funcionalidad a una clase dinámicamente. En cada archivo layout.php se define el código HTML común de cada vista, evitando que este sea repetido en cada página.

2.5 Diagramas necesarios para la implementación.

Representación de los diagramas que hacen posible la implementación del sistema. Entre los diagramas encontramos, diagrama de componentes, diagrama de clases del diseño y diagrama de despliegue.

2.5.1 Diagrama de componentes.

El diagrama de componentes perteneciente al módulo Perfil del STI, implementa la arquitectura MVC. En la Presentación están los componentes JavaScript, css y php necesarios para asegurar la vista al cliente, estos componentes tendrán una ubicación específica en el directorio del proyecto para una mejor gestión de su información. En la lógica del negocio se encuentran las clases.php que controlan los eventos a partir de lo capturado en la vista, de igual forma se comunica con el modelo y este último con la BD mediante el framework Propel que incluye el framework Symfony en su arquitectura.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

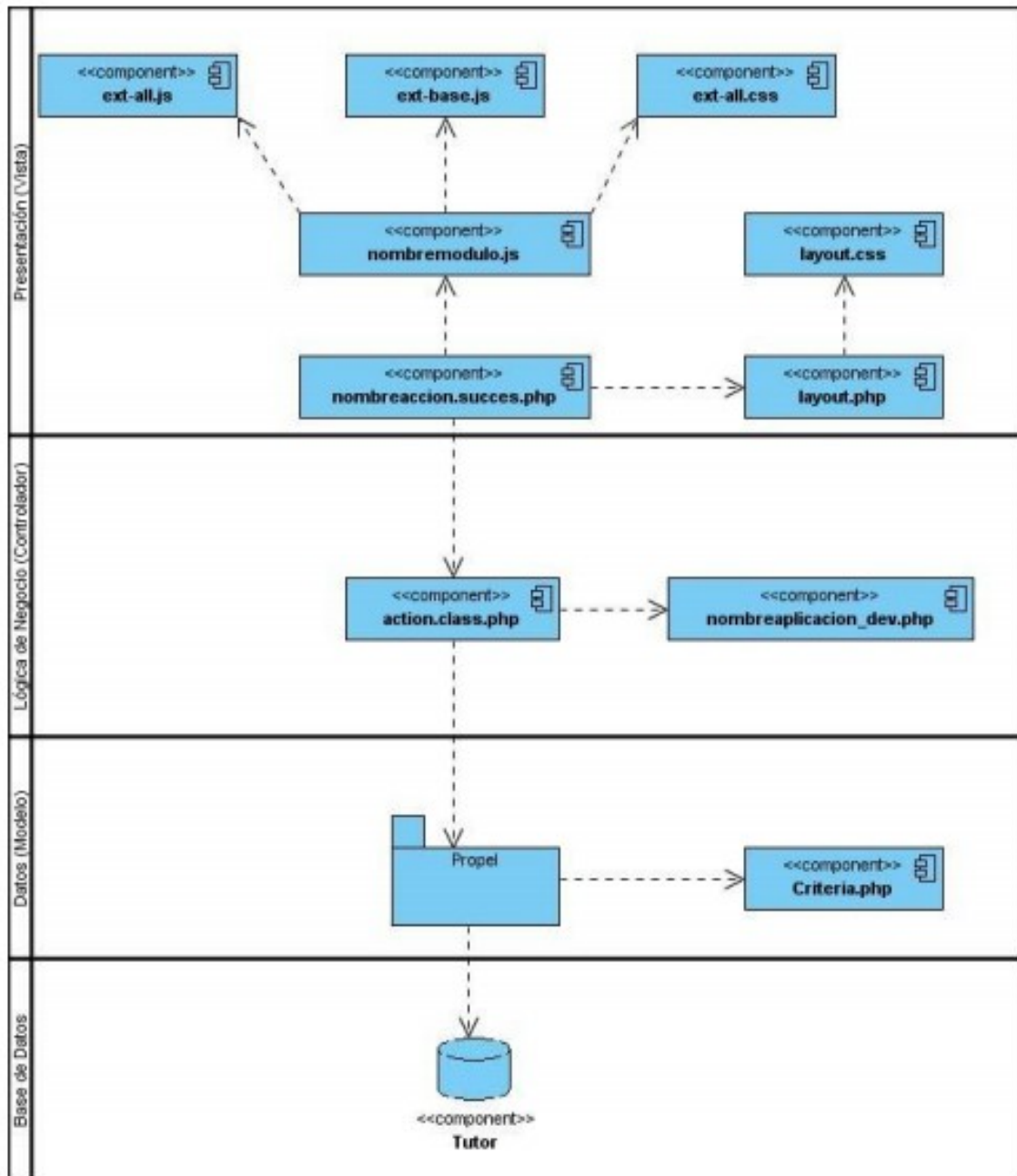


Figura 11: Diagrama de componentes del módulo Perfil.

2.5.2 Diagrama de clases del diseño.

El diagrama de clases del diseño Perfil de Usuario muestra la relación entre la clase controladora PerfilAction, que se relaciona con la vista a través del archivo indexSuccess, y con las clases entidades del modelo; de esta forma se evidencia el uso de la arquitectura MVC definida por el framework Symfony.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

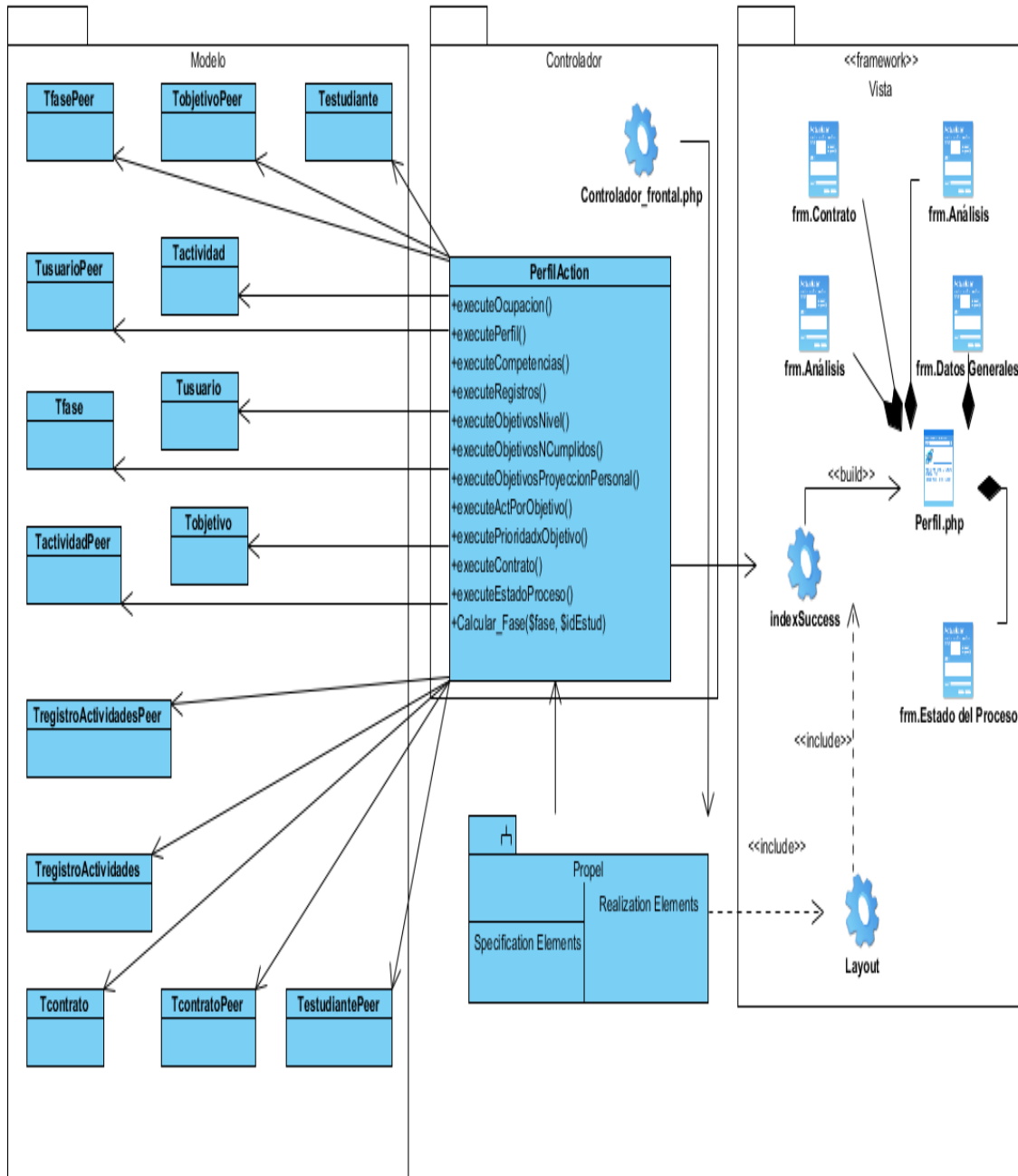


Figura 12: Diagrama de clases del diseño del módulo Perfil.

2.5.3 Diagrama de despliegue.

El usuario, desde una estación de trabajo y haciendo uso de un navegador web, podrá acceder al sistema el cual estará desplegado en un servidor web donde se encontrará ubicado el STI VIRTEVALL. Dicha aplicación estará conectada a un servidor de Bases de Datos en el cual se almacenará la información de interés para la solución.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA



Figura 13: Diagrama de despliegue del módulo Perfil.

2.6 Estándar de codificación.

El código desordenado es difícil de leer y las personas pueden perder interés si no son capaces de descifrar lo que el código intenta hacer. Resulta importante que los desarrolladores puedan entender el código rápidamente, lo que permite realizar modificaciones y mejoras en un periodo de tiempo reducido. El código es una forma de comunicación, y así como a alguien podría no querer leer un libro con errores ortográficos y mala puntuación, los programadores deben intentar escribir buen código de tal forma que sea fácil de entender y modificar por otros. El estilo de programación se refiere a la forma en que se da formato al código fuente. Esto involucra la forma en que se ubican las llaves, se indenta el código y se utilizan los paréntesis, el uso de salto de líneas.

La elaboración y el empleo de estándares de codificación ofrecen numerosas ventajas a la hora de elaborar cualquier tipo de producto software y entre sus principales ventajas podemos mencionar:

- ✓ Se reduce la posibilidad de cometer errores.
- ✓ Se obtiene un código legible y comprensible.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

- ✓ Mejora la comunicación entre los miembros del grupo de programadores del equipo de desarrollo.

Para la elaboración de un estándar de codificación se debe tener en cuenta, entre otros aspectos, las especificaciones y características del framework de desarrollo y el o los lenguajes de programación escogidos.

Los aspectos para los que generalmente se establecen estándares son los siguientes:

- ✓ Identificadores.
- ✓ Indentación.
- ✓ Líneas y espacios en blanco.
- ✓ Comentarios.

2.6.1 Identificadores.

En el caso de los identificadores existen estilos definidos mundialmente como el lowerCamelCase y el UperCamelCase. Cada palabra interna en identificadores compuestos comienza con mayúsculas para ambos estilos, además ocurre que no se colocan caracteres de separación entre las palabras que conforman un identificador compuesto en ninguno de los dos casos. Para el primero, el identificador comienza con minúscula y para el segundo, el identificador comienza con mayúscula.

- ✓ Espacio de nombre: para nombrar los espacios de nombre se definió el UperCamelCase.
- ✓ Clase: para nombrar las clases se definió el UperCamelCase.

```
class PerfilACTIONS extends sfActions {  
    <CÓDIGO DE LA CLASE ...>  
}
```

- ✓ Variable: para nombrar las variables se definió el lowerCamelCase.

```
private $idAlumno;  
private $objetivoEstudianteFase;
```

- ✓ Función: para nombrar las funciones se definió el UperCamelCase.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

```
function executeEstadoProceso(){
    <CÓDIGO DE LA FUNCIÓN ...>
}
```

2.6.2 Indentación.

Por indentación se entiende mover un bloque de texto hacia la derecha insertando espacios o tabuladores para separarlo del texto adyacente. Para dar un ejemplo, a continuación se muestra un código no indentado, y uno correctamente indentado:

```
function executeEstadoProceso(){
    $F = Array();
    $X = Array();
    $X[0] = 0; $X[1] = 0;
    $N = 0;
    if($usuario->getFase()%2 == 0){
        $F[] = PerfilActions::Calcular_Fase();
        $F[] = PerfilActions::Calcular_Fase();
        $X[0] = TfasePeer::obteneridFase();
        $X[1] = TfasePeer::obteneridFase();
    }
    else{
        $F[] = PerfilActions::Calcular_Fase();
        $X[0] = TfasePeer::obteneridFase();
    }
}
```

El mismo código, pero aplicando indentación:

```
function executeEstadoProceso(){
    $F = Array();
    $X = Array();
    $X[0] = 0; $X[1] = 0;
    $N = 0;
    if($usuario->getFase()%2 == 0){
        $F[] = PerfilActions::Calcular_Fase();
        $F[] = PerfilActions::Calcular_Fase();
        $X[0] = TfasePeer::obteneridFase();
        $X[1] = TfasePeer::obteneridFase();
    }
    else{
        $F[] = PerfilActions::Calcular_Fase();
        $X[0] = TfasePeer::obteneridFase();
    }
}
```

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

En la implementación del sistema se define como estilo de indentación el **K&R** y **BSD KNF**. El estilo K&R es el más usado en el lenguaje C y PHP. Se trata de abrir la llave en la misma línea de declaración de la orden, indentando los siguientes pasos al mismo nivel que la llave y cerrando la llave en el mismo nivel que la declaración. Normalmente las tabulaciones en Windows son de 4 espacios, cuando las tabulaciones tienen 8 espacios se trata del estilo BSD o KNF. Se entiende mejor con el siguiente ejemplo:

```
function executeEstadoProceso(){
    if($usuario->getFase()%2 == 0){
        $F[] = PerfilActions::Calcular_Fase();
    }
    else{
        $F[] = PerfilActions::Calcular_Fase();
    }
}
```

La ventaja de usar este estilo es que las llaves iniciales no necesitan ninguna línea extra para ellas solas, por lo que se ahorra espacio vertical de lectura.

2.6.3 Líneas y espacios en blanco.

Para mejorar la legibilidad y organización del código muchas veces se utilizan líneas en blanco para separar segmentos de código que pueden corresponder a clases, funciones, declaraciones, implementaciones, comentarios, bloques o sencillamente secciones críticas que se deseen despejar. Así mismo sucede con los espacios en blanco cuando se utilizan para separar elementos dentro de las sentencias de código. En ocasiones se separan con espacios cada operador de su respectivo operando, paréntesis, identificadores, símbolos y algunos lenguajes exigen que se separen las palabras propias del vocabulario de las adyacentes para ser comprendidas por los compiladores. En este trabajo se ha definido emplear líneas en blanco:

- ✓ Entre funciones.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

```
function executeEstadoProceso(){
    <CÓDIGO DE LA FUNCION ...>
}

function executeCalcularFase($fase, $id_estud){
    <CÓDIGO DE LA FUNCION ...>
}
```

- ✓ Entre las palabras reservadas y los elementos adyacentes a las mismas.

```
private $idAlumno;
private $objetivoEstudianteFase;
```

- ✓ Después de cada punto y coma (;) en las estructuras for.

```
for($i = 0; $i < count($objetivos_vencidos); $i++){
    <CÓDIGO DE LA INSTRUCCIÓN ...>
}
```

2.6.4 Comentarios.

El uso de comentarios durante la codificación ha demostrado que es beneficiosa por varias razones:

- ✓ Ayuda al programador a entender cada elemento o sección de código.
- ✓ Hace más fácil el proceso de adaptación del código durante su reutilización.
- ✓ Sirve de guía en los casos en que varios programadores trabajen sobre las mismas secciones del código.
- ✓ Disminuye el esfuerzo de análisis ya que el lenguaje natural es más legible que cualquier lenguaje de programación.

Todo esto se aprecia claramente cuando se escribe gran cantidad de código en largos intervalos de tiempo donde generalmente el o los programadores olvidan lo que se pensó en un momento. Los comentarios se pueden utilizar para varios fines:

- ✓ Para explicar el propósito de las funciones.
- ✓ Para explicar las características fundamentales de las clases.
- ✓ Para sintetizar las acciones de los algoritmos complejos.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

- ✓ Para aclarar los datos que representan las variables.
- ✓ Para dividir secciones de código en dependencia de los diferentes contextos y funciones.
- ✓ A veces se usan comentarios temporales para recordar cosas que faltan, cosas que se deben modificar o analizar en otro momento.

Es necesario tener en cuenta algunos detalles al escribir comentarios:

- ✓ La capacidad de síntesis.
- ✓ El uso de lenguaje técnico.
- ✓ No repetir exactamente paso por paso lo que hace el algoritmo sino expresar un resumen de su propósito.
- ✓ Usar un estilo uniforme de comentario definido en estándares para todo el equipo.
- ✓ Escribir el comentario solo donde sea necesario.

En este trabajo se decidió colocar los comentarios encima de la línea a la que se le quiera aplicar y encima de la línea cabecera de los bloques. La indentación se hará al nivel de la línea en cuestión. Se utilizarán en funciones y clases. Se puede utilizar en algoritmos no triviales y secciones de diferentes contextos dentro de los métodos.

2.7 Integración mediante el perfil de usuario.

Para mostrar la integración mediante el perfil de usuario se hace necesario el esclarecimiento de las etapas por las que pasa el estudiante cuando se enfrenta al sistema por primera vez o continuamente, las cuales se describen a continuación:

- ✓ **La etapa diagnóstica:** El estudiante ya autenticado previamente en el sistema realiza los diagnósticos generales de la personalidad y el diagnóstico de posicionamiento de nivel de idioma. Estos diagnósticos tiene como objetivo diagnosticar las características personales de los estudiantes y el nivel de idioma. Estas características son las estrategias de aprendizaje, estilos de aprendizaje, y el perfil de inteligencia.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

La realización de estos diagnósticos se divide en dos fases fundamentales. La primera fase consiste en la realización de encuestas para determinar la clasificación del estudiante en visual, verbal-auditivo, kinestésico o indeterminado de acuerdo a sus estilos de aprendizaje. Luego de realizadas las encuestas se le presenta al estudiante el diagnóstico de nivel de idioma personalizado para determinar el nivel de idioma que dará paso a la negociación. (Ver figura 14).

Otra de las etapas por las que pasa el estudiante es:

- ✓ **La etapa de negociación:** En este punto el estudiante ya pasó por la etapa diagnóstica y es el momento de la presentación del perfil del estudiante y aprobación del mismo por parte del estudiante. El objetivo de esta etapa es que el estudiante se identifique con la caracterización de su personalidad y con el nivel de idioma obtenido a partir de los diagnósticos realizados en la etapa anterior.

Esta etapa es un punto crítico para el estudiante y para la personalización de la evaluación a través del Perfil de Usuario, pues esas son las evidencias bases de las cuales el Perfil de Usuario se nutre para lograr una integración con los demás módulos. En caso de que el estudiante no se sienta satisfecho con los resultados mostrados siempre puede volver a la fase anterior y comenzar el proceso de nuevo.

En la siguiente fase es ya donde el estudiante interactúa con las actividades evaluativas que lo van a entrenar en un nivel de idioma.

- ✓ **La etapa de entrenamiento:** El objetivo de esta fase es adiestrar al estudiante en la realización de actividades evaluativas que le ayuden a vencer los objetivos del nivel de idioma en que se encuentra.

El perfil de usuario con ayuda de los resultados de los diagnósticos ya realizados comienza a personalizar el proceso de evaluación del aprendizaje a través de un tratamiento diferenciado a los estudiantes, en correspondencia con los estilos de aprendizaje, nivel de idioma, tipo de usuario y nivel de entrenamiento que poseen los mismos. Todo esto se hace con el objetivo de

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

que el estudiante desarrolle sus potencialidades en consecución de los objetivos del nivel e idioma en que se encuentra. Todo ello se hace posible mediante el tratamiento de las evidencias dejadas por el estudiante a medida que realiza actividades e interactúa con el sistema.

La última etapa y por ello no significa que sea menos importante en el proceso de evaluación del aprendizaje autónomo es:

- ✓ **La etapa de certificación:** Es esta fase donde el estudiante hace la solicitud para avalar el nivel de idioma desarrollado.

Una vez que el estudiante ha transitado por las etapas anteriores solicita a través de su perfil de estudiante la realización de un examen para certificar el nivel de idioma. En el perfil de usuario se gestiona la solicitud y envío de la solicitud de certificación de nivel de idioma a la persona responsable de analizarla para su aprobación o rechazo. Esta persona analiza si existen suficientes evidencias del desarrollo de las habilidades en idioma inglés del estudiante y aprueba o no que se realice el examen. Estas evidencias son mostradas al profesor a través del perfil del estudiante, al cual el profesor tiene acceso.

Se ha descrito el perfil de usuario de manera que evidencia la integración de los demás módulos del sistema, personalizando el proceso de evaluación del aprendizaje de idioma inglés.

CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Menú Estudiante

- Mi perfil
- Encuestas
- Diagnósticos
- Actividades
- Evaluaciones
- Calendario
- Mi Diario

Calendario

Abril 2012

L	M	M	J	V	S	D
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Hoy

Datos Generales | Registro Actividades | Análisis | Contrato | Estado del Proceso

1

Nombre: Adrian Gonzalez Ruiz	Nivel de Auto Evaluacion: 1
Ocupacion: Alumno	Tipo de Usuario: Visual
Facultad: 15	Nivel de Entrenamiento: Básico
Grupo: 15511	Fase: A2
Año: 5	Nivel Acreditado: Realizar Exámen de Acreditación

2

Información General:

En Ud. predomina el Canal Visual de percepción de la información. Le presta más atención a la información que recibe visualmente y le resulta más fácil reconstruir en la mente la información visual. O sea, le es más fácil representar visualmente lo que sabe. Ud. entiende mejor las órdenes si se le dice y le son mostradas. Considera detalles y es muy refinado en la presentación escrita de materiales. Su producción oral tiende a ser más limitada. Utiliza repertorios de palabras que evidencian su preferencia por la información que llega por un canal visual.

Usted utiliza la estrategia de aprendizaje ENSAYO de forma predominante, lo que indica que usted es un aprendiz mayormente reflexivo e independiente y con alto énfasis en recursos memorísticos como principal soporte en su aprendizaje. Tiene además predilección por el canal de aprendizaje visual. Gusta de utilizar la repetición como método de aprendizaje.

Ud tiene predominio de la inteligencia lógica-matemática, que es la capacidad de razonamiento lógico, que se utiliza para resolver problemas de lógica y matemáticas. Esta asociada a las habilidades de comprender y resolver cálculos numéricos, problemas de lógica y conceptos abstractos. Es la inteligencia desarrollada en todas las disciplinas científicas.

3

Figura 14: Integración del módulo Diagnóstico mediante el Perfil de Usuario.

2.8 Conclusiones

En este capítulo se realizó un análisis del diseño propuesto por la tesis de Análisis y Diseño que antecede a esta investigación. Se hizo un esbozo de los componentes utilizados en la solución de las funcionalidades principales. Se logró establecer una línea base sencilla que permita la programación eficiente por parte del desarrollador. Se establecieron los estándares de codificación pertinentes para que el código generado tenga la legibilidad necesaria al trabajar con algoritmos críticos y grandes volúmenes de código. Se propusieron ejemplos sencillos incluyendo figuras para su comprensión. Se analizó la integración de los demás módulos del sistema mediante el Perfil de Usuario.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Capítulo 3: Validación de la solución propuesta.

3. Introducción.

En este capítulo se hace una búsqueda de las técnicas de prueba de software que contribuyan comprobar la validez de la solución obtenida. Se detallan los casos de prueba con sus objetivos y el alcance de cada uno de ellos. Se aborda el tema relacionado con las herramientas empleadas en el proceso de pruebas y se analizan los resultados obtenidos.

3.1 Pruebas de software.

Para comprobar el grado de cumplimiento de los requisitos obtenidos en la primera fase se usan las pruebas del sistema. Estas definen un conjunto amplio de acciones de comprobación que abarcan todas las características que determinan la calidad de un software. Se comprueban las funcionalidades del sistema diseñando casos de prueba que definen cómo proceder. Estos casos de prueba incluyen los juegos de datos a usar que son los válidos o esperados, y los no válidos o no esperados por el programa. Además, se establecen los resultados a alcanzar en correspondencia de la lógica del programa y los datos ingresados. Se describen las condiciones generales en las que se debe aplicar las pruebas para obtener los objetivos propuestos. El objetivo de los casos de prueba es forzar al máximo el sistema en los puntos críticos para encontrar fallos y detectar defectos. Las pruebas se deben aplicar durante todo el ciclo de vida del software e invariablemente se le debe dedicar una gran parte del esfuerzo total del desarrollo. Se deben planificar correctamente desde el inicio y establecer qué hacer, cómo hacer, quién va a hacer y en qué condiciones hacer las comprobaciones. Es beneficioso que los desarrolladores prueben su producto pero que no falte la mano de terceras personas que no intervinieron en el proyecto directamente, pero que tengan conocimientos sobre aspectos de calidad de software, ya que así se detecta mayor cantidad de fallas. (Pressman. R, 2002)

Se deben escoger los tipos de prueba que se adapten mejor al sistema que se va a probar. La selección de los tipos de prueba deben estar en correspondencia con el lenguaje de programación, el proceso de desarrollo, las

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

características de los desarrolladores, el tipo de funcionalidad que se implementa, la plataforma en que se ejecutan los procesos, los errores más importantes, si el sistema es de escritorio o web, si realiza conexiones a Bases de Datos o no.

3.2 Automatización de pruebas.

En el desarrollo de aplicaciones web el proceso de probar el sistema de una manera correcta supone un gran esfuerzo. Esto está dado por el hecho de que los requisitos del sistema están sujetos a cambios constantes, lo que implica un gran número de versiones del sistema y con ello la aparición de nuevos errores. Este es el motivo por el que la automatización de pruebas es una recomendación, aunque no una obligación, útil para crear un entorno de desarrollo satisfactorio. Las pruebas automatizadas permiten garantizar que los cambios no introducen incompatibilidades en el funcionamiento del sistema. Además, este tipo de pruebas obligan a los programadores a crear pruebas en un formato estandarizado y muy rígido que pueda ser procesado por un framework de pruebas. (Potencier. F y otros, 2009)

El tener el proceso de pruebas automatizado permite ilustrar el funcionamiento del sistema, y al realizar un gran número de estas se puede mostrar la salida que produce el sistema para una serie de entradas de pruebas, brindando la posibilidad de entender el propósito de cada método. El proceso de automatización de pruebas es la parte del ciclo de calidad en la que el software de automatización es utilizado para controlar la ejecución de pruebas, comparación de resultados, preparación de precondiciones y realización de informes. Los dos grandes grupos de pruebas unitarias existentes son las pruebas de Caja Negra y las pruebas de Caja Blanca.

3.3 Herramientas para la realización de las pruebas.

En el ámbito de PHP existe diversidad de marcos de trabajo para crear pruebas unitarias y funcionales, siendo los más conocidos PHPUnit y SimpleTest. Symfony incluye su propio marco de trabajo, cuyo nombre es Lime. Se basa en la biblioteca Test::More de Perl y es compatible con TAP. Esto significa que los resultados de las pruebas se muestran con el formato definido en el "*Test*

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Anything Protocol", creado para facilitar la lectura de los resultados de las pruebas. (Potencier. F y otros, 2009)

El Lime presenta además una serie de ventajas dentro de las cuales resaltan por su importancia las siguientes: (Potencier. F y otros, 2009)

- ✓ Ejecuta los archivos de prueba en un entorno independiente para evitar interferencias entre las diferentes pruebas. No todos los marcos de trabajo de pruebas garantizan un entorno de ejecución "limpio" para cada prueba.
- ✓ Las pruebas de Lime son fáciles de leer y sus resultados también lo son. En los sistemas operativos que lo soportan, los resultados de Lime utilizan diferentes colores para mostrar de forma clara la información más importante.
- ✓ Symfony utiliza Lime para sus propias pruebas y su "regression testing", por lo que el código fuente de Symfony incluye muchos ejemplos reales de pruebas unitarias y funcionales.
- ✓ Está escrito con PHP, es muy rápido y está bien diseñado internamente. Consta únicamente de un archivo, llamado lime.php, y no tiene ninguna dependencia.

3.4 Pruebas de Caja Negra

Las Pruebas de Caja Negra deben su nombre a los elementos que estas revisan y las condiciones en que se hace la revisión. Estas se basan en los requisitos funcionales del sistema y se llevan a cabo desde el exterior de la aplicación. Este tipo de prueba es importante a la hora de medir el grado de cumplimiento de los requisitos solicitados por el cliente y se aplican sobre la interfaz de la aplicación observando las respuestas del sistema ante determinadas acciones, y los datos de salida para determinados datos de entrada. (Pressman. R, 2002)

El método seleccionado para la realización de las pruebas fue la Partición Equivalente. Pressman (2002) presenta la Partición Equivalente como un método de prueba de Caja Negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

3.4.1 Descripción del caso de prueba proyección personal.

El usuario introduce los datos de su proyección personal, en caso de dejar algún campo vacío necesario que la proyección personal demande, el sistema muestra un error, de lo contrario la introduce correctamente y lo notifica.

Condiciones de Ejecución.

- ✓ El usuario debe estar identificado y autenticado en el sistema.
- ✓ El usuario debe haber realizado el diagnóstico de posicionamiento que sitúa al usuario en un nivel y una fase de idioma.
- ✓ El usuario debe seleccionar la opción Análisis.
- ✓ Finalmente el usuario selecciona la opción guardar proyección personal.

Nombre	Escenarios	Descripción	Flujo Central
Proyección Personal.	1.1 Adicionar proyección personal.	El CU comienza cuando el usuario accede a la opción Análisis del perfil de usuario.	<ol style="list-style-type: none"> 1. El sistema muestra la interfaz que permite adicionar la proyección personal. 2. El usuario introduce los datos de su proyección personal. 3. El sistema verifica que todos los campos se hallan llenado por parte del usuario. 4. Si todos los campos fueron llenados correctamente el sistema inserta la proyección personal en la BD y termina el CU.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

	1.2 Datos incompletos.	Detectar que faltan datos por llenar o insertar.	1. El sistema detecta que existen campos por llenar y muestra un error al usuario.
--	------------------------	--	--

Tabla 1: Escenarios a probar en el CU Proyección Personal.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	de Observación.
El usuario entra datos válidos para adicionar su proyección personal.		El sistema muestra un mensaje de confirmación indicando que se ha realizado la operación satisfactoriamente.	El sistema muestra el mensaje: "Datos Guardados".	
	El usuario no entra todos los datos de su proyección personal.	El sistema muestra un mensaje de error.	El sistema muestra el mensaje: "Verifique sus Datos".	

Tabla 2: Caso de prueba Proyección Personal.

3.4.2 Descripción del caso de prueba Contrato.

El usuario introduce los datos de su contrato, en caso de dejar algún campo vacío necesario que el contrato demande, el sistema muestra un error, de lo contrario los introduce correctamente y lo notifica.

Condiciones de Ejecución.

- ✓ El usuario debe estar identificado y autenticado en el sistema.
- ✓ El Usuario debe haber realizado su proyección personal.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

- ✓ El usuario debe seleccionar la opción Contrato.
- ✓ Finalmente el usuario selecciona la opción realizar contrato.

Nombre	Escenarios	Descripción	Flujo Central
Contrato	1.1 Realizar Contrato.	El usuario comienza cuando accede a la opción Contrato del perfil de usuario.	<p>CU</p> <p>5. El sistema muestra la interfaz que permite adicionar el contrato.</p> <p>6. El usuario introduce los datos de su contrato</p> <p>7. El sistema verifica que todos los campos se hallan llenado por parte del usuario.</p> <p>8. Si todos los campos fueron llenados correctamente el sistema inserta el contrato en la BD y termina el CU.</p>
	1.2 Datos incompletos.	Detectar que faltan datos por llenar o insertar.	2. El sistema detecta que existen campos por llenar y muestra un error al usuario.

Tabla 3: Escenarios a probar en el CU Contrato.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observación.
El usuario entra datos válidos para adicionar su contrato.		El sistema muestra un mensaje de confirmación indicando que se ha realizado la operación satisfactoriamente.	El sistema muestra el mensaje: "Contrato Aceptado."	
	El usuario no entra todos los datos de su proyección personal.	El sistema muestra un mensaje de error.	El sistema muestra el mensaje: "No debe dejar ningún campo en blanco. Todos son requeridos".	

Tabla 4: Caso de prueba Contrato.

3.5 Pruebas de Caja Blanca.

Las pruebas de Caja Blanca se nombran de esta forma porque a diferencia de las pruebas de Caja Negra que actúan sobre la interfaz, estas revisan la parte interna del software, específicamente sobre el código fuente. En estas pruebas se comprueban los caminos lógicos del sistema, generando casos de prueba que ejerciten las estructuras condicionales y los bucles, es por ello que las pruebas unitarias se basan en las Técnicas de Pruebas de Caja Blanca. Existen varios métodos que analizan diferentes partes del programa, uno de ellos es la técnica del Camino Básico que se utiliza para comprobar la complejidad lógica de un diseño procedimental, permite diseñar casos de

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

prueba para cubrir todas las sentencias de un programa a partir de la obtención de un conjunto de caminos independientes. Un resultado fundamental de estas pruebas es la complejidad ciclomática, esta acota la cantidad mínima de casos de prueba que se deben ejecutar.

3.5.1 Diseño de casos de prueba de Caja Blanca aplicados.

Consecutivamente se muestra la prueba del camino básico aplicada a un método de la clase PerfilActions del módulo Perfil de Usuario, para la cual se siguieron una serie de pasos lógicos.

- ✓ Se construye el grafo de flujo a partir del código fuente del método a probar.
- ✓ Se determina la complejidad ciclomática $V(G)$ del grafo G . Para calcular la complejidad ciclomática hay tres formas:
 - $V(G) = \text{Aristas} - \text{Nodos} + 2$.
 - $V(G) = \text{Nodos de predicado} + 1$.
 - $V(G) = \text{Número de regiones del grafo}$.
- ✓ A partir del valor de la complejidad ciclomática obtenemos el número de caminos independientes, que nos dan un valor límite para el número de pruebas que tenemos que diseñar.

Descripción del método EstadoProceso.

Este método es el encargado de obtener el nivel de idioma desarrollado por el estudiante y mostrárselo, valida que los datos obtenidos sean correctos.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

```
function executeEstadoProceso(){
    $F = Array();
    $X = Array();
    $X[0] = 0; $X[1] = 0;
    $usuario = TestudiantePeer::ObtenerAlumnoId($this->getUser()->getAttribute('id_usuario_activo'));
    $N = 0;
    if($usuario->getFase()%2 == 0){
        $F[] = PerfilActions::CalcularFase($usuario->getFase()-1, $usuario->getIdAlumno());
        $F[] = PerfilActions::CalcularFase($usuario->getFase(), $usuario->getIdAlumno());
        $X[0] = TfasePeer::ObteneridFase($usuario->getFase()-1)->getPrioridad();
        $X[1] = TfasePeer::ObteneridFase($usuario->getFase())->getPrioridad();
    }
    else{
        $F[] = PerfilActions::CalcularFase($usuario->getFase(), $usuario->getIdAlumno());
        $X[0] = TfasePeer::ObteneridFase($usuario->getFase())->getPrioridad();
    }

    for($i = 0; $i < count($F);$i++){
        $N = $N + $F[$i] * $X[$i];
    }
    $this->renderText(json_encode($N));
    return sfView::NONE;
}
```

Figura 15: Función EstadoProceso de la clase PerfilActions.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Complejidad Ciclomática	
$V(G) = \text{Aristas} - \text{Nodos} + 2$	
$V(G) = (7 - 6) + 2$	
$V(G) = 3.$	
Posibles Caminos	
1-2-4-5-4-6	
1-3-4-5-4-6	

Tabla 5: Grafo de Flujo.

3.5.2 Casos de pruebas.

A cada camino obtenido de la prueba de camino básico se le realiza un caso de prueba. Se considera importante aclarar que en estos casos de prueba se verifica que el sistema devuelva un número contenido entre los valores 0 y 1, todo depende de la cantidad de objetivos vencidos que tenga el estudiante por habilidad.

3.5.2.1 Caso de prueba para el camino básico número uno.

Descripción: Los datos se obtienen a través del método Post, conteniendo el id del usuario activo.

```
103 $juego_datos_1 = '38';
```

Figura 16: Juego de datos para el caso de prueba uno.

Condición de ejecución: El usuario debe estar autenticado en el sistema y debe haber realizado el Diagnóstico de Posicionamiento.

Obtiene: Se obtuvo el resultado esperado para el juego de datos 1.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Resultados esperados: Se actualiza y se le muestra al estudiante en la pestaña Estado del Proceso del Perfil el % de cumplimiento del nivel en que se encuentra.

Resultado arrojado a través de la herramienta Lime que permite verificar cada uno de los caminos básicos de una función mediante los juegos de datos prefijados.

```
E:\xampp\htdocs\Uirtevall_1.7.1.1>symfony test-unit Estado
1..3
ok 1 - Prueba satisfactoria, juego de datos 1
  Looks like you planned 3 tests but only ran 1.

E:\xampp\htdocs\Uirtevall_1.7.1.1>_
```

Figura 17: Resultado de la herramienta Lime para el caso de prueba uno.

3.5.2.2 Caso de prueba para el camino básico número dos.

Descripción: Los datos se obtienen a través del método Post, conteniendo el id del usuario activo.

```
106 $juego_datos_2 = '237';
```

Figura 18: Juego de datos para el caso de prueba dos.

Condición de ejecución: El usuario debe estar autenticado en el sistema y debe haber realizado el Diagnóstico de Posicionamiento.

Obtiene: Se obtuvo el resultado esperado para el juego de datos 2.

Resultados esperados: Se actualiza y se le muestra al estudiante en la pestaña Estado del Proceso del Perfil el % de cumplimiento del nivel en que se encuentra.

Resultado arrojado a través de la herramienta Lime que permite verificar cada uno de los caminos básicos de una función mediante los juegos de datos prefijados.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

```
E:\xampp\htdocs\Uirtevall_1.7.1.1>symfony test-unit Estado
1..3
ok 1 - Prueba satisfactoria, juego de datos 2
Looks like you planned 3 tests but only ran 1.
E:\xampp\htdocs\Uirtevall_1.7.1.1>_
```

Figura 19: Resultado de la herramienta Lime para el caso de prueba dos.

3.6 Resumen de las Pruebas.

Las pruebas de Caja Negra permitieron validar las especificaciones en términos de requisitos funcionales por parte del cliente, las mismas encontraron a través de los casos de pruebas seleccionados 20 no conformidades que en posteriores iteraciones fueron resueltas. Las cuales se evidencian en el siguiente gráfico.

Las pruebas de Caja Blanca mediante su principal resultado: la complejidad ciclomática, permitió diseñar casos de prueba para cubrir todas las sentencias de una función a partir de la obtención de un conjunto de caminos independientes.

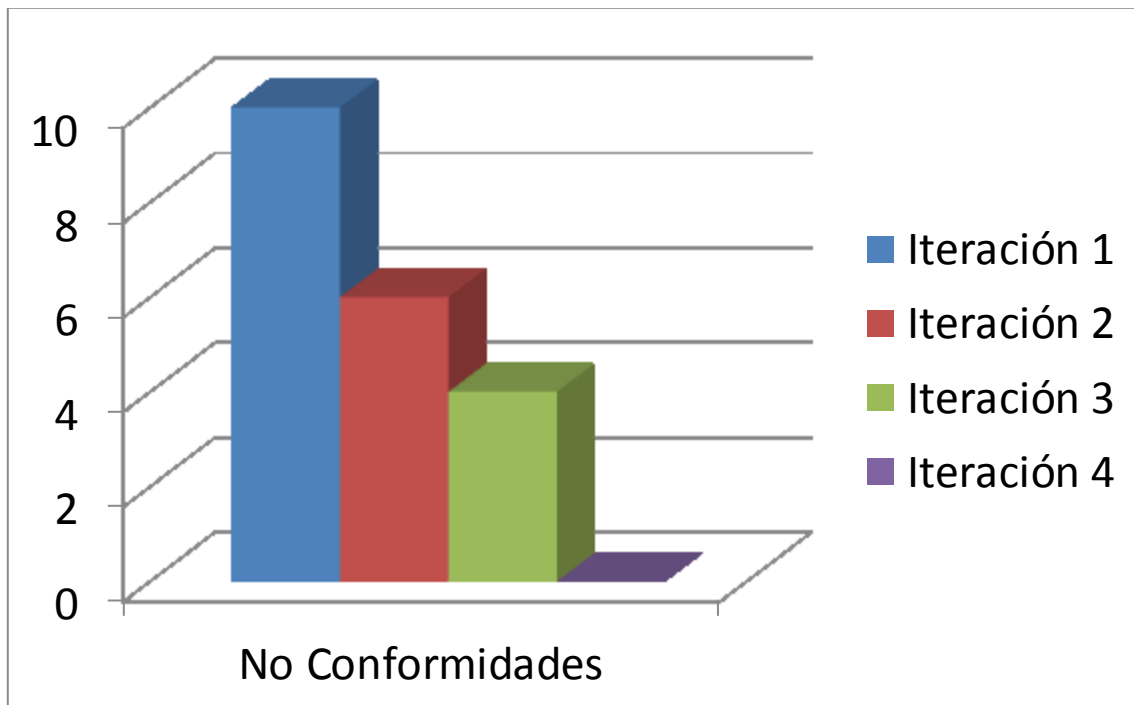


Figura 20: Gráfico de las Pruebas de Caja Negra.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

3.7 Conclusiones.

En este capítulo se lleva a cabo el proceso de pruebas para verificar si el sistema presenta o no deficiencias, utilizando el método de caja negra y caja blanca. Destacar además, que durante el proceso de implementación del módulo Perfil de Usuario se han llevado a cabo pruebas de aceptación por parte de su cliente. La constante interacción con el cliente contribuyó a que durante el proceso de pruebas los resultados obtenidos fueran satisfactorios.

CONCLUSIONES

Conclusiones.

A modo de conclusiones se puede afirmar que los principales resultados con el desarrollo de la presente investigación son los siguientes:

- ✓ El estudio de los antecedentes de los STI en el mundo, en Cuba y en la UCI, contribuyó a trazar una línea para el desarrollo del módulo, cuya arquitectura, luego de ser analizada, permitió personalizar la toma de decisiones a partir de las evidencias dejadas por el usuario.
- ✓ La técnica de la IA aplicada: "*Redes Bayesianas*", se utilizó con el fin de evaluar el nivel de idioma en el diagnóstico, el entrenamiento y la certificación del aprendizaje autónomo del usuario.
- ✓ Se caracterizó el empleo de estándares de codificación para los identificadores, líneas y espacios en blancos, comentarios e indentación. Esto permitió que el código fuera más legible y fácil de comprender para futuras versiones del sistema.
- ✓ Se implementó el módulo Perfil de Usuario del Sistema Tutor Virtual para la Evaluación del Aprendizaje Autónomo de Idiomas Extranjeros.
- ✓ Se modelaron y ejecutaron varios casos de pruebas, específicamente la Partición Equivalente para las pruebas de Caja Negra y las pruebas de Camino Básico para las pruebas de Caja Blanca, que validaron el funcionamiento de la aplicación desarrollada.

Se puede concluir que se cumplió satisfactoriamente el objetivo general enunciado desde el inicio de la investigación práctica, por lo que se le dio solución al problema científico planteado.

BIBLIOGRAFÍA

Recomendaciones.

Con vistas a dar continuidad al desarrollo de este proyecto se recomienda:

- ✓ Implementar un sistema de notificaciones que permita mantener al tanto al estudiante del cumplimiento o no de las tareas propuestas, según el tiempo definido por el usuario para cada una de ellas.

BIBLIOGRAFÍA

Bibliografía

Abud. A, 2009. Metodología de Ingeniería de Software Educativo. Instituto Tecnológico de Orizaba. México.

Ariza. E y otros, 2004. Uso de un sistema tutorial inteligente en el ámbito educativo. Universidad Autónoma Metropolitana de Xochimilco. México.

Badaracco. M, 2008. Enfoque para el Desarrollo de un Sistema Tutor Inteligente Basado en Competencias. Universidad de Málaga. España.

Badaracco. M, 2010. Arquitectura para un Sistema Tutor Inteligente basado en Competencias (STI-C). Evaluación empírica simulada. XVI Congreso Argentino de Ciencias de la Computación. Argentina.

Cataldi. Z y otros, 2005. Sistemas Tutores inteligentes: Los estilos del estudiante para selección del tutorizado. Facultad de Ingeniería de la Universidad de Buenos Aires. Argentina.

Cataldi. Z y otros, 2007. Diseño de Sistemas Tutores Inteligentes con Tecnología de Agentes: Los Agentes Docentes en el Módulo Tutor. Facultad de Ingeniería de la Universidad de Buenos Aires. Argentina.

Cataldi. Z y otros, 2007. Fundamentos para el Submódulo Evaluador en Sistemas Tutores Inteligentes: Diagnóstico, predicción y autoevaluación. XIII Congreso Argentino de Ciencias de la Computación. Argentina.

Cataldi. Z y otros, 2007. Modelado del Estudiante en Sistemas Tutores Inteligentes. Facultad de Ingeniería de la Universidad de Buenos Aires. Argentina.

Cataldi. Z y otros, 2009. Sistemas Tutores Inteligentes orientados a la enseñanza para la comprensión. Facultad de Ingeniería de la Universidad de Buenos Aires. Argentina.

Caviedes. D y otros, 2009. Diseño de un sistema tutor inteligente basado en estilos cognitivos. Universidad Distrital Francisco José de Caldas. Colombia.

BIBLIOGRAFÍA

Ecured, 2011. Sencha ExtJs. [Citado el: 11 de Diciembre de 2011.]. http://www.ecured.cu/index.php/Sencha_Ext_JS.

Esquivias. J y otros, 2005. Bases para la construcción de un Sistema Tutor Inteligente para el aprendizaje de la citología cervico-vaginal. VII Congreso Virtual Hispanoamericano de Anatomía Patológica.

E.U.I.T.I.O. Entornos de Desarrollo. Sitio Web oficial de la E.U. Ingeniería Técnica Informática de Oviedo. [Citado el: 12 de Dicimebre de 2011.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.

Falkenauer. E, 1999. Evolutionary Algorithms: Applying Genetic Algorithms to Real-World. New York. Springer. United States of America.

Fernando. S, 2005. Sistemas Inteligntes para el Modelado del Tutor. Universidad de Buenos Aires. Argentina.

Fernández. P y otros, 2009. Un Tutor Inteligente para la Visualización de Métodos Algorítmicos y Estructuras de Datos. Facultad de Informática Universidad Complutense de Madrid. España.

Fernando. S y otros, 2005. Nuevo enfoque metodológico para el diseño de de los sistemas tutores inteligentes a partir de un acercamiento distribuido. Laboratorio de Sistemas Inteligentes, Facultad de Ingeniería, Universidad de Buenos Aires. Argentina.

Gardner. H, 1993. Inteligencias Múltiples: La teoría en la práctica. Barcelona Paidós. España.

Gardner. H, 2001. La inteligencia reformulada: las inteligencias múltiples en el siglo XXI. Barcelona Paidós. España.

Giraffa. L y otros, 1997. Multi-Ecological: an Learning Environment. Universidade de Coimbra. Portugal.

Haykin. S, 1999. Neural Networks: A comprehensive foundation. Prentice Hall. United States of America.

BIBLIOGRAFÍA

Huapaya. C y otros, 2007. Modelo de evaluación del conocimiento en un Sistema Tutorial Inteligente. Facultad de Ingeniería, Universidad Nacional de Mardel Plata. Argentina.

Jiménez. M, 2009. Modelo de Evaluación Adaptativa del Nivel de Conocimiento del Estudiante para Sistemas Tutores Inteligentes. Universidad Nacional de Colombia. Colombia.

Kohonen. T, 1998. An introduction to neural computing. Neural networks.

Martínez. Y, 2008. Evaluación del aprendizaje autónomo para entorno virtual del inglés en la Universidad de las Ciencias Informáticas. 6to Congreso Internacional de Educación Superior. 978-959-282-069-2.

Martínez. Y, 2009. Evaluación del aprendizaje autónomo para entorno virtual del aprendizaje del inglés en la UCI. 6to Taller Internacional Innovación Educativa-Siglo XXI. 978-959-16-1045-4.

Martínez. Y, 2009. Evaluación del aprendizaje autónomo para entorno virtual del aprendizaje del inglés en la UCI. 7th latin American and Caribbean Conference for Engineering and Technology. 0-9822896-2-6.

Martínez. Y, 2011. Conceptualización de una Metodología de Evaluación Integradora del Aprendizaje Autónomo de Idiomas Extranjeros. XIV Congreso Internacional de Informática Educativa en la XIV Convención y Feria Internacional. 978-959-7213-01-7.

Martínez. Y, 2011. La evaluación integral del aprendizaje autónomo de idioma Inglés con el uso de un Sistema Tutor Inteligente. Holguín: V Conferencia Científica Internacional. 978-959-16-1329-5.

Mayet. M, 2010. Algoritmos de IA aplicados a un Sistema Tutor Inteligente para enseñar lengua inglesa. Conferencia Científica de la Universidad de las Ciencias Informáticas. Cuba

NetBeans. NetBeans Community. NetBeans. [Citado el: 12 de Diciembre de 2011.] <http://netbeans.org/>. <http://netbeans.org/>.

BIBLIOGRAFÍA

Pressman. R, 2002. Ingeniería de Software. Un enfoque práctico. McGraw-Hill. México.

PostgreSQL. PostgreSQL. PostgreSQL. [Citado el: 11 de Diciembre de 2011.] <http://www.postgresql.org/>.

Potencier. F y otros, 2009. Symfony 1.2, la guía definitiva. http://www.librosweb.es/symfony_1_2.

Quiñones. E, 2006. Introducción a PostgreSQL. Sitio sobre PostgreSQL. http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf.

Raquel. C, 2009. Sistemas Tutoriales Inteligentes. Un análisis crítico. Universidad Nacional de La Plata. Argentina.

Rodríguez. O, 2010. Arquitectura del Tutor Virtual de Evaluación para el Aprendizaje Autónomo de Idiomas (VIRTEVALL). Universidad de las Ciencias Infomáticas. Cuba.

Rojas. A, 2007. Sistema Tutorial Inteligente para el desarrollo de habilidades en la solución de problemas matemáticos. Facultad de Ciencias de la Educación, Universidad de La Salle. Colombia.

Sierra. E y otros, 2004. Fundamentos para una metodología de diseño de Sistemas Tutoriales Inteligentes centrada en la reparación de mecanismos. Universidad de Buenos Aires. Argentina.

Sierra. E y otros, 2007. Líneas de investigación del grupo de Sistemas Inteligentes aplicados a Ingeniería. Universidad Nacional del Comahue. Argentina.

Scheuermann. F y otros, 2009. The Transition to Computer-Based Assessment. European Communities.

Soledad. C, 2004. Sistemas Inteligentes en la Educación: Una revisión de las líneas de investigación actuales. Revista Electrónica de Investigación y Evaluación Educativa.

BIBLIOGRAFÍA

Sykes. E y otros, 2003. A Prototype for an Intelligent Tutoring System for Students Learning to Program in Java (TM). School of Computing and Information Management. Canada.

Urretavizcaya. M, 2001. Sistemas Inteligentes en el ámbito de la Educación.

Vanlehn. K, 1988. Student Modelling. M. Polson. Foundations of Intelligent Tutoring systems. Hillsdale. N.J. Lawrence Erlbaum Associates.

Wolf. B, 1984. Context Dependent Planning in a Machine Tutor. Dissertation University of Massachusetts. United States of America.