

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 7



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

Sistema para el montaje de un tutorial inteligente.

Autor(es): Edney Nápoles Mejías.
Rosalba Rivera Peguero.

Tutor(es): Dayana Caridad Tejera Hernández.
Danay Hernández León.

La Habana, Junio 2013
"Año 55 de la Revolución"



Declaramos ser los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ___ días del mes de _____ del año 2013.

Edney Nápoles Mejías.
(Firma del autor)

Rosalba Rivera Peguero.
(Firma del autor)

Msc. Dayana Caridad Tejera Hernández.
(Firma del tutor)

Ing. Danay Hernández León.
(Firma del tutor)

Msc. Dayana Caridad Tejera Hernández.

Graduada de Ingeniería en Ciencias Informáticas en 2007. Se desempeña actualmente como asesora del Departamento Metodológico Central de Ingeniería y Gestión de Software (IGSW) de la UCI y como profesora principal de las asignaturas Ingeniería de Software 1 y 2 (IS1 e IS2). En 7 años de labor docente (incluidos 2 años de alumna ayudante), ha impartido las asignaturas IS1, IS2, Gestión de Software, Metodología de la Investigación Científica e Historia de la Informática. Se desempeñó como jefa del departamento de IGSW y PP y luego como asesora de calidad en la facultad 3. Más adelante fue coordinadora de 5to año en la Dirección de Planificación de la Vicerrectoría de Formación. Posee categoría docente principal de Profesor Asistente.

Correo electrónico: dtejera@uci.cu

Ing. Danay Hernández León

Graduada de Ingeniería en Ciencias Informáticas en 2008. Inició su trabajo en la UCI en el año 2009 como Tercerizado del Dpto. Despliegue Soporte y Atención al Cliente, realizando los diagnósticos alasPACS en los hospitales: Hermanos Ameijeiras, Pando Ferrer, CIMEQ, entre otros. Además, se mantuvo trabajando en el tema de Despliegue, Soporte y Capacitación en el Centro de Investigaciones Médicas Quirúrgicas CIMEQ, manteniéndose al frente del desarrollo de la actividad en dicha institución, priorizando y asegurando la presencia en el hospital del personal de la UCI. Logró mantener actualizado el sistema instalado, asegurando así el buen funcionamiento y puesta en marcha de la solución alas PACS-RIS. Realizó un proceso de centralización de la información de los sistemas desplegados o en proceso piloto del centro de salud (CESIM), entre los que se encuentra alasPACS, alasRIS y Balance Material, con el objetivo de documentar todas las acciones que se realizan en el centro vinculado directamente con los temas de despliegue de los productos. En el curso 2011-2012 llevó el control de los activos fijos del CESIM obteniendo un buen resultado en las auditorías realizadas al Centro. Participó en la creación del Consejo Editorial de la Facultad y de la Universidad, estableció el grupo editorial de la facultad definiendo las políticas editoriales internas, el flujo de información y aprobación de los artículos a publicar. Actualmente imparte clases de Ingeniería de Software en la facultad, funge como profesora guía del grupo 7302, hasta el momento está obteniendo buenos resultados en la labor que se encuentra realizando.

Correo electrónico: dhleon@uci.cu

De Edney:

Dedico esta tesis a mi hermanita que con solo tres añitos ha sabido llenar de alegría el corazón en nuestra familia y es para ella que me voy a graduar. Te quiero mucho.

A mi madre por estar siempre ahí cuando más lo necesitaba, por ayudarme y apoyarme para pasar estos cinco años de la carrera sin ningún problema.

A mi padre por ayudarme también en todo momento y por estar ahí también para apoyarme.

A mis abuelos, los cuatro en general porque han aportado su grano de arena para que yo llegara a la meta en la universidad y siempre darme todo para estar aquí.

Los quiero mucho a todos.

De Rosalba:

A mi mamá y a mi hermano, por todo su esfuerzo porque yo sea alguien en la vida, nadie ha recibido más afecto y amor incondicional como el que ustedes me han dado.

De Edney y Rosalba:

A los profesores, por brindar su ayuda, conocimiento y apoyo.

A las tutoras: Dayana y Danay, por guiar nuestras ideas, no solo en el desarrollo de esta tesis, sino también en nuestra formación como investigadores.

A todos aquellos que de una forma u otra han contribuido a nuestra formación como personas y como profesionales a lo largo de estos años.

A todas aquellas personas con las que hemos compartido, que hacen que la vida tenga sentido, por brindarnos su apoyo y comprensión.

De Edney:

Le quiero agradecer a mi madre, a mi padre, a mis abuelos, a mis tíos por apoyarme en todo momento y estar ahí para mí.

A mis compañeros de aula, aquellos que preguntaron cómo saliste en el corte, como vas en la tesis y en especial agradecerle a Yanet que siempre me soportó y mira que hay que soportarme, pero la quiero mucho esa es la hermana de acá de la universidad.

A Itamys por todos los momentos agradables que he pasado junto a ella por sus carcajadas pesadas pero bobas, una como ella no voy a conocer.

A Raúl ese hermano que aunque ya se graduó siempre me prestó su ayuda en la tesis, me brindó su casa, me dio comida en la semana de la victoria, y fue un placer conocerlo y haber convivido cuatro años en la universidad, nuestra amistad no se acaba acá sino que se agranda aún más, gracias Raúl y a Sandra, por ser una persona maravillosa grandemente un placer haberte conocido.

A otra tía que conocí en la isla y mira que me dice cuando voy para allá para que esté con ella nada más, claro siempre hay sus celos por ahí pero te mando un beso Ania.

Quiero agradecer grandemente a Adrián por prestarme su ayuda en la tesis y revisar cada detalle de concordancia en los párrafos y decirme que estaban feos, pero los arreglaba.

A todos los profesores maravillosos que he conocido en la universidad y me han brindado su apoyo para que la tesis saliera bien y con una nota excelente, a Zoila, mis tutoras Dayana y Danay muchas gracias.

A Rosalba por ser una compañera de tesis excelente y déjenme decirles, hay quien decía por ahí yo no voy a hacer la tesis contigo, pero nada fue maravilloso este poco tiempo trabajando juntos.

Agradecerle a Ángel Omelio que nos prestó su ayuda grandemente en la tesis, se reía de nuestros disparates, nos decía están fritos, pero nos enseñó muchas cosas y siempre estuvo ahí para cualquier duda o situación que se presentara.

Agradecer a todas las personas que he conocido y que aportaron su granito de arena, Yelena gracias por prestarme tu cuenta porque la mía se terminaba y yo te gastaba la tuya, a Liliana, Dailín, Yolexis, Naryara, Lisandra, Odelis, Marihelem, Liana, Michy muchas gracias fue maravilloso compartir con ustedes.

De Rosalba:

Primero que nadie quiero agradecerle a mi mamá Ada, por apoyarme y ayudarme siempre que lo necesite y soportarme en mis momentos de estrés, gracias por ser más que una madre una amiga y hacerme saber a cada momento que siempre puedo contar contigo, gracias por tu amor, dedicación, comprensión y paciencia, todo lo que soy es por ti y te lo debo a ti.

A mi hermanito Roilán, many eres la luz de mis ojos, como digo yo (el amor de mi vida, "el verdadero") gracias por quererme tanto, por existir, por mostrar tu preocupación por mis cosas, por aconsejarme y quiero que sepas que estoy y estaré siempre muy orgullosa de ti.

A todos mis tíos y tías que siempre he podido contar con ellos y han estado al pendiente de mí y de mi carrera, en especial a mi tía favorita, tía mama, gracias por creer siempre en mí, aun cuando ni yo lo hacía, siempre me decías "tu si puedes " y mira, pude, gracias por escucharme, comprenderme, apoyarme y ser tan buena amiga.

A las amigas que hice en la universidad Heily, Yami, Marianni, Dairis y Dailín por apoyarme y brindarme su ayuda siempre que lo necesite.

A la más grande de todas Sissi por estar siempre conmigo en los momentos buenos y en los malos, soportarme día a día y mostrarme su apoyo incondicional durante los 5 años de la carrera, acompañarme a cada fiesta, aunque no fueron muchas.

A mi grupo en general 7503 que de una forma u otra siempre estuvieron ahí para mortificar y ayudar.

A Ariel un muy buen amigo que siempre se encargó de hacer más alegres los problemas y situaciones, hacerme ver el lado bueno de lo que sucedía y por su amistad incondicional y sincera.

A las amistades que no son de la universidad pero no menos importantes a Yesenia, una muy buena amiga por escucharme cada fin de semana y preguntarme siempre como me iba con las pruebas , a Lalo que llamaba y se preocupaba por los chismes.

A mi compañero de tesis Edney por tener toda la paciencia del mundo conmigo, entenderme y apoyarme, por convertirse en un gran amigo que a pesar de sus pesadeces y decir No Rosalba siempre pude contar con él.

A Alejandro que me brindó su ayuda y su amor en todos los momentos que lo necesité y en los que no también, sus palabras siempre fueron muy reconfortantes, cuando lo sentía todo perdido decía, Tu veras que vas a salir bien.

A los profesores de la facultad que me formaron durante cinco años y en especial a las tutoras Dayana y Danay por corregirme y guiarme para una buena investigación y exposición.

A Ángel Omelio que nos ayudó mucho y estuvo pendiente de cada corte y del resultado obtenido, por atendernos y siempre con muy buen carácter.

A todas las personas que he conocido durante la universidad que me han ayudado de alguna forma y han estado presente en algún momento de mi vida.



Cuando fui a la escuela, me preguntaron que quería ser de mayor. Yo respondí: "feliz". Me dijeron que yo no entendía la pregunta, y yo les respondí que ellos no entendían la vida.

La vida es lo que sucede mientras estás ocupado haciendo otros planes.

John Lennon.

Resumen

La introducción y uso de las Tecnologías de la información y las comunicaciones (TIC) ha logrado un salto vertiginoso en el desarrollo científico técnico a escala mundial. Estas han transformado la manera de trabajar y gestionar recursos en las organizaciones. Actualmente la Universidad de las Ciencias Informáticas (UCI), se encuentra inmersa en la búsqueda de nuevas formas y tecnologías educativas que permitan facilitar el proceso de enseñanza de los estudiantes, teniendo en cuenta el estilo de aprendizaje de los mismos, para proporcionar la ayuda pedagógica adecuada.

El objetivo concreto del presente trabajo, es el desarrollo de un Sistema Tutorial Inteligente (STI) de apoyo al Proceso de Enseñanza Aprendizaje (PEA) de la asignatura Ingeniería de Software 2 para la Facultad 7, que ayude a resolver las tareas docentes en los temas de la asignatura, utilizando el Razonamiento Basado en Casos (RBC) como técnica de Inteligencia Artificial (IA). Se obtendrá como resultado un sistema web para ser utilizado por estudiantes y profesores.

Palabras claves: proceso de enseñanza aprendizaje, razonamiento basado en casos, sistema tutorial inteligente, tecnología educativa.

Índice de contenido

Resumen.....	IX
Introducción.....	1
Capítulo 1: Fundamentación Teórica	4
1. Introducción.	4
1.1 El proceso de enseñanza aprendizaje en el UCI. Análisis de los problemas en la asignatura Ingeniería de Software 2.....	4
1.1.1 La educación, el estudiante y el profesor.	5
1.1.2 Proceso de enseñanza aprendizaje. Elementos.	5
1.2 Software educativo. Medio de enseñanza aprendizaje.	7
1.2.1 Características del software educativo.	7
1.3 Inteligencia artificial.....	7
1.3.1 Técnicas de inteligencia artificial.....	8
1.4 Sistemas tutoriales inteligentes.	8
1.4.1 Arquitectura de los STI.	9
1.4.2 Los sistemas de aprendizaje basado en casos. Su utilización en el desarrollo de tutoriales inteligentes.....	10
1.5 Análisis de soluciones existentes.	11
1.5.1 Soluciones existentes a nivel internacional.	11
1.5.2 Soluciones existentes a nivel nacional.	12
1.5.3 Valoración de las herramientas.....	12
1.6 Metodologías de desarrollo de software.....	13
1.6.1 Selección de la metodología a utilizar.	16
1.7 Lenguaje Unificado de modelado. UML 2.1.	17
1.8 Herramienta CASE.....	17
1.8.1 Visual Paradigm. 6.4	17
1.8.2 Rational Rose.	18
1.8.3 Selección de la herramienta CASE.....	18
1.9 Lenguajes de programación.	19
1.9.1 Selección del lenguaje de programación a utilizar.	20
1.10 Sistemas Gestores de Base de Datos.....	20

1.10.1 Selección del Sistema Gestor de Base de Datos.....	21
Consideraciones parciales.....	22
Capítulo 2. Descripción del sistema.....	23
2. Introducción.....	23
2.1 Descripción del sistema. Propuesta de solución.....	23
2.1.1 Resultados esperados del sistema.....	23
2.2 Personas relacionadas con el sistema.....	25
2.3 Descripción del dominio.....	25
2.4 Lista de reserva del producto.	26
2.4.1 Requisitos funcionales.	26
2.4.2 Requisitos no funcionales.	27
2.5 Planificación del proyecto.	29
2.5.1 Historias de usuarios.....	29
2.6 Plan de iteraciones.....	30
2.7 Plan de entrega.....	32
Consideraciones parciales.....	33
Capítulo 3. Diseño, Implementación y Pruebas.....	34
3. Introducción.....	34
3.1 Diseño del sistema.....	34
3.2 Tarjetas CRC. Cargo o Clase, Responsabilidad y Colaboración.	34
3.3 Patrón de arquitectura y diseño.....	38
3.3.1 Descripción del patrón MVC.....	39
3.4 Modelo de datos.....	42
3.5 Implementación del sistema.	43
3.5.1 Tareas de la implementación.	44
3.6 Diagrama de despliegue.	44
3.7 Pruebas.....	45
3.7.1 Pruebas de aceptación.....	45
3.7.2 Resultados de las pruebas de aceptación.	45
3.8 Pruebas realizadas a los estudiantes.....	45
3.8.1 Verificación de la consistencia de los datos.	46
3.8.2 Resultado de la prueba del software.	46

Consideraciones parciales.....	50
Conclusiones generales.....	51
Recomendaciones.....	52
Bibliografía y Referencias bibliográficas.....	53
Anexos.....	59
Anexos 1.....	59
Anexos 2.....	66
Anexos 3.....	84
Anexos 4.....	87

Índice de tablas

Tabla 1. Tarjeta CRC clase estudiante.	35
Tabla 2. Tarjeta CRC clase evaluación.	36
Tabla 3. Tarjeta CRC clase encuesta satisfacción.	36
Tabla 4. Tarjeta CRC clase tarea.	36
Tabla 5. Tarjeta CRC clase material tarea.	37
Tabla 6. Tarjeta CRC clase asignatura.	37
Tabla 7. Tarjeta CRC clase tema.	37
Tabla 8. Tarjeta CRC clase habilidad.	38
Tabla 9. Tarjeta CRC clase habilidad específica.	38
Tabla 10. Tarjeta CRC clase material educativo.	38
Tabla 11. HU_1 Autenticar usuario.	66
Tabla 12. HU_2 Realizar encuesta de estilo aprendizaje.	67
Tabla 13. HU_3 Registrar estudiante.	67
Tabla 14. HU_4 Ver listado estudiantes.	68
Tabla 15. HU_5 Gestionar tarea de estudiantes.	69
Tabla 16. HU_6 Asignar materiales para resolver tareas.	70
Tabla 17. HU_7 Ver tareas.	71
Tabla 18. HU_8 Subir tareas resueltas.	72
Tabla 19. HU_9 Evaluar tareas.	73
Tabla 20. HU_10 Mostrar evaluación de tareas.	73
Tabla 21. HU_11 Gestionar material educativo.	74
Tabla 22. HU_12 Ver material educativo.	75
Tabla 23. HU_13 Realizar encuesta de satisfacción.	76
Tabla 24. HU_14 Mostrar resultado encuesta de satisfacción.	77
Tabla 25. HU_15 Gestionar asignatura.	78
Tabla 26. HU_16 Gestionar tema.	78
Tabla 27. HU_17 Gestionar habilidades.	79
Tabla 28. HU_18 Gestionar habilidades específicas.	80
Tabla 29. HU_19 Gestionar material para las habilidades específicas.	81
Tabla 30. HU_20 Relacionar habilidad tarea.	82
Tabla 31. HU_21 Mostrar tareas resueltas adjuntadas.	83
Tabla 32. HU_22 Eliminar estudiante.	83
Tabla 33. TI_1 Describir el sistema propuesto y creación del modelo de dominio.	84
Tabla 34. TI_2 Diseñar base de datos.	85
Tabla 35. TI_3 Describir los requisitos funcionales del sistema.	85
Tabla 36. TI_4 Confeccionar historias de usuario.	86
Tabla 37. TI_5 Generar las clases modelo.	86
Tabla 38. TI_6 Diseñar las interfaces de usuario.	86
Tabla 39. TI_7 Implementar las funcionalidades descritas en las historias de usuario.	87

Tabla 40. Prueba de aceptación PA_1 para historia de usuario HU_1.....	87
Tabla 41. Prueba de aceptación PA_2 para historia de usuario HU_2.....	88
Tabla 42. Prueba de aceptación PA_3 para historia de usuario HU_3.....	88
Tabla 43. Prueba de aceptación PA_4 para historia de usuario HU_4.....	89
Tabla 44. Prueba de aceptación PA_5 para historia de usuario HU_5.....	89
Tabla 45. Prueba de aceptación PA_6 para historia de usuario HU_6.....	90
Tabla 46. Prueba de aceptación PA_7 para historia de usuario HU_7.....	90
Tabla 47. Prueba de aceptación PA_8 para historia de usuario HU_8.....	91
Tabla 48. Prueba de aceptación PA_9 para historia de usuario HU_9.....	91
Tabla 49. Prueba de aceptación PA_10 para historia de usuario HU_10.....	92
Tabla 50. Prueba de aceptación PA_11 para historia de usuario HU_11.....	92
Tabla 51. Prueba de aceptación PA_12 para historia de usuario HU_12.....	93
Tabla 52. Prueba de aceptación PA_13 para historia de usuario HU_13.....	93
Tabla 53. Prueba de aceptación PA_14 para historia de usuario HU_14.....	94
Tabla 54. Prueba de aceptación PA_15 para historia de usuario HU_15.....	94
Tabla 55. Prueba de aceptación PA_16 para historia de usuario HU_16.....	95
Tabla 56. Prueba de aceptación PA_17 para historia de usuario HU_17.....	95
Tabla 57. Prueba de aceptación PA_18 para historia de usuario HU_18.....	96
Tabla 58. Prueba de aceptación PA_19 para historia de usuario HU_19.....	96
Tabla 59. Prueba de aceptación PA_20 para historia de usuario HU_20.....	97
Tabla 60. Prueba de aceptación PA_21 para historia de usuario HU_21.....	97
Tabla 61. Prueba de aceptación PA_22 para historia de usuario HU_22.....	98

Introducción

La introducción y uso de las Tecnologías de la Información y las Comunicaciones (TIC), ha logrado un salto vertiginoso en el desarrollo científico técnico a escala mundial. Estas han transformado la manera de trabajar y gestionar recursos en las organizaciones (1). La informatización de la sociedad en Cuba, se considera como el proceso de utilización ordenada y masiva de las TIC para satisfacer las necesidades de información y conocimiento de todas las personas, esferas de la sociedad y facilitar el Proceso de Enseñanza Aprendizaje (PEA). (2)

La enseñanza en el ambiente computacional se emplea como una solución al crecimiento exponencial del conocimiento de la personas, lo que conlleva a la búsqueda de nuevas soluciones en la pedagogía y las tecnologías educativas de enseñanza y aprendizaje. En este caso están los sistemas basados en la Inteligencia Artificial (IA) o inteligentes, con el objetivo de construir modelos para facilitar y agilizar el desempeño educativo. Debido a esto, tanto profesores, como estudiantes, deben familiarizarse con dichas tecnologías, de forma tal que puedan aplicarlas para mejorar el PEA.

Mediante lo expuesto anteriormente surge en septiembre de 2002 la Universidad de las ciencias Informáticas. En la UCI se desarrollan diferentes alternativas que hacen incursionar a los estudiantes en el trabajo con tecnologías educativas. El modelo docente implementado en la universidad, que integra la formación, la producción y la investigación, se auxilia de un sistema de gestión del aprendizaje basado en Moodle, más comúnmente conocido como Entorno Virtual de Enseñanza Aprendizaje (EVEA). En este interactúan estudiantes y profesores, y se mantiene organizada toda la información referente a las diversas asignaturas del plan de estudio de la carrera. Sin embargo, en el EVEA existen algunos recursos que permiten personalizar el PEA, pero no tienen en cuenta las preferencias de los estudiantes para el estudio. (3)

Los problemas que provocan baja motivación de los estudiantes en la universidad para acceder al EVEA, se evidencian en que los cursos, estudios independientes, actividades, conferencias y clases prácticas, no se adaptan a su forma de abstraer los conocimientos. Los materiales referentes a la asignatura Ingeniería de Software 2 no están bien organizados y la falta de información provoca que la asignatura sea difícil de comprender por los alumnos indistintamente de su nivel y forma de aprendizaje. Un indicador claro de esto es la escasa participación de estudiantes en exámenes de premio de las correspondientes asignaturas Ingeniería de Software 1 y 2. Los alumnos por su parte señalan que las tareas les resultan

tediosas, no se sienten motivados pues los medios y métodos con que se imparten estas asignaturas no siempre les atraen.

El escaso trabajo intencionado con los alumnos de alto aprovechamiento reflejado en los colectivos de estas asignaturas y lo anteriormente expresado provoca que los estudiantes no posean las habilidades básicas ni el contenido bien consolidado para resolver las tareas orientadas. La falta de experiencia del claustro para enfrentar las diferencias individuales de cada estudiante, evidencia que en el desarrollo de las conferencias y las clases prácticas no enfatizan en la forma de percibir la información y procesar los conocimientos. Debido al modelo docente implementado, no tienen el tiempo suficiente para ejercitar el contenido de las diversas asignaturas del plan de estudio de la carrera, pues el profesor no está siempre disponible para aclararle las dudas.

A partir de lo expuesto anteriormente se define como **problema a resolver**: ¿Cómo personalizar las tareas docentes teniendo en cuenta las características de los estudiantes en la asignatura Ingeniería de Software 2, para facilitar la independencia del profesor?

Este problema está enmarcado en el **objeto de estudio** referido al desarrollo de software educativo para el apoyo del PEA. El **campo de acción** a partir del objeto de estudio, comprende la construcción de un tutorial inteligente de apoyo al PEA en la asignatura Ingeniería de Software 2.

Como **objetivo general**, para brindar solución al problema planteado se ha definido: Desarrollar un sistema tutorial inteligente que permita la personalización de las tareas docentes teniendo en cuenta las características de los estudiantes para la asignatura Ingeniería de Software 2.

Por tal razón se plantea como **idea a defender**: El desarrollo de un sistema de apoyo al PEA en la asignatura Ingeniería de Software 2 que permita la resolución de las tareas orientadas y la consultoría de material complementario, influye de forma positiva en el PEA de los estudiantes.

Tareas a desarrollar para dar cumplimiento al objetivo trazado son:

1. Análisis del estado del arte de los Sistemas Tutoriales Inteligentes (STI) existentes a nivel internacional y nacional.
2. Selección de las herramientas, tecnologías y lenguajes para el desarrollo del sistema.
3. Elaboración de la estructura de la base de datos con PostgreSQL.
4. Desarrollo del sistema para la creación de un tutorial inteligente.

5. Prueba del sistema con una muestra de estudiantes.

Al desarrollar estas tareas, se obtiene un Sistema Tutorial Inteligente de apoyo al PEA de la asignatura Ingeniería de Software 2, los **aportes y beneficios** esperados son los siguientes:

- ✓ Asegurará la personalización de las tareas docentes en la asignatura Ingeniería de software 2, a través del estilo de aprendizaje de los estudiantes registrados en el sistema.
- ✓ Proveerá al estudiante de los medios necesarios para resolver las tareas docentes según sus características, basado en la forma de percibir la información y procesar los conocimientos.
- ✓ Permitirá guardar información sobre la satisfacción de los estudiantes respecto a su trabajo con la aplicación, y ayudará al estudio sistemático sin la dependencia del profesor.
- ✓ La herramienta posibilitará su uso y funcionamiento en las asignaturas del plan de estudio de la carrera.

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

Capítulo 1 - Fundamentación teórica: el tema tratado finalmente no es más que el objeto de estudio, el campo de acción y la situación problemática presente, así como la selección de la plataforma, lenguaje, herramientas y metodología con las que se llevará a cabo la propuesta de solución.

Capítulo 2 – Descripción del sistema: se muestran las características del software con las etapas de desarrollo, propias de la metodología seleccionada para guiar el proceso, obteniéndose los artefactos de cada etapa.

Capítulo 3 – Diseño, Implementación y Pruebas: se abordan las etapas de diseño, implementación y pruebas, con las cuales se finaliza el proceso de desarrollo del software y por tanto la propuesta del sistema que presenta este trabajo.

Capítulo 4 – Validación del sistema: se aborda la verificación y validación del sistema a través de pruebas realizadas al software y se verifica la consistencia de la información insertada en la base de datos para comprobar correctamente su funcionamiento.

Capítulo 1: Fundamentación Teórica.

1. Introducción.

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

1.1 El proceso de enseñanza aprendizaje en el UCI. Análisis de los problemas en la asignatura Ingeniería de Software 2.

La misión de la educación superior cubana es: Preservar, desarrollar y promover la cultura de la humanidad, a través de sus procesos sustantivos, en plena integración con la sociedad; llegando con ella a todos los ciudadanos, con pertinencia y calidad y contribuir así al desarrollo sostenible del país (4).

En la Universidad de las Ciencias Informáticas (UCI) se han dado pasos de avance, perfeccionando el plan de estudio y cada una de las asignaturas, prestando mayor atención al papel de los colectivos de año, el profesor guía y el tutor, fortaleciendo el trabajo de los colectivos de asignatura en la formación de valores desde lo curricular, dándole su justa importancia al diagnóstico pedagógico en la caracterización de los estudiantes (5). Sin embargo aún es insuficiente el aprovechamiento de los resultados de este diagnóstico en el trabajo con las individualidades de cada estudiante, en todas estas instancias. En los informes semestrales de las asignaturas, siguen señalándose como deficiencia este aspecto. (6)

Un indicador claro de esto es la escasa participación de estudiantes en exámenes de premio de las asignaturas Ingeniería de software 1 y 2. De igual forma se refleja como una constante en los resultados de los controles a clase realizados: la falta de experiencia del claustro para enfrentar las diferencias individuales de los estudiantes, las dificultades en el diseño de las tareas docentes a desarrollar en las clases y de las que se orientan como parte del estudio individual o por equipos fuera del aula. Los estudiantes por su parte, señalan que las tareas les resultan tediosas, no se sienten motivados pues los medios y métodos con que se imparten estas asignaturas no siempre les atraen. (7)

1.1.1 La educación, el estudiante y el profesor.

La educación, vista desde un sentido amplio, constituye un proceso orientado al desarrollo personal, donde el educando, simultáneamente, construye conocimientos y se desarrolla en planos diversos como persona. El PEA que se desarrolla para lograrlo, es esencialmente interactivo y comunicativo, y en él interviene la subjetividad del que enseña y del que aprende, dada por sus motivaciones, aspiraciones, conflictos, necesidades y cualidades personales. De ellas, las menos trabajadas son las dos últimas; se aprecia bastante poco la intencionalidad del profesorado en la adaptación del PEA a cada estudiante según sus características y sus necesidades al aprender. (8)

El estilo de aprendizaje permite centrar a los estudiantes en el PEA y personalizar dicho proceso, pero es particularmente concreta la definición en que el estilo son las formas relativamente estables de las personas para aprender, a través de las cuales se expresa el carácter único e irrepetible de la personalidad, la unidad de lo cognitivo y lo afectivo, y entre otras, sus preferencias al percibir y procesar la información, al organizar el tiempo y al orientarse en sus relaciones interpersonales durante el aprendizaje. (9)

Lo anteriormente expresado evidencia que todos los estudiantes no poseen las mismas características y habilidades para entender los conocimientos que brinda el profesor cuando imparte las conferencias. Este por su parte debe estar pendiente y conocer el avance y desempeño que va teniendo el alumno para ser capaz de corregir los problemas y dificultades que pueda presentar, tener en cuenta que cada aprendiz puede asimilar los contenidos de forma diferente y proponer vías, métodos que le faciliten entender los contenidos, siendo esto una meta en la educación actual.

1.1.2 Proceso de enseñanza aprendizaje. Elementos.

Se entiende por PEA como el sistema de comunicación intencional que se produce en un marco institucional en el que se generan estrategias encaminadas a provocar el aprendizaje. (10)

El aula es el lugar en el cual se reúnen los cuatro elementos claves del PEA: el alumno, el docente, la información que se va a comunicar y el medio (11). En la figura 1 se muestra esta relación.



Figura 1. Relación entre los elementos del aprendizaje.

El alumno, es poseedor de estrategias de aprendizaje adquiridas durante su vida académica, es poseedor de un perfil social y emocional que lo diferencia de sus compañeros, además es el elemento central del PEA. El docente, es poseedor del conocimiento y de las estrategias pedagógicas, es el motivador, administrador de la información y de los elementos que integran el medio que rodea al alumno. La información está delimitada por los planes y programas de estudio. (11)

El medio, está constituido por la infraestructura como: laboratorios, talleres, herramientas informáticas, medios de comunicación, recursos del aula, y otras herramientas que están a disposición de los alumnos para facilitar el aprendizaje y para mejorar la actividad social entre ellos. El medio en el cual se ve envuelto el aprendizaje, ha ido evolucionando a lo largo del tiempo, lo que ha redefinido el rol del docente, este nuevo rol le permite ser el administrador del conocimiento, el facilitador del aprendizaje, y el motivador que trabaja constantemente sobre las expectativas de los alumnos. (12)

De acuerdo con lo expuesto, se considera que el acto de enseñar y aprender no debe denominarse como principios para la enseñanza, sino que debe incluir al alumno y permitirle al educador dirigir científicamente el desarrollo integral de la personalidad de estos, considerando sus estilos de aprendizaje en medios propicios para la comunicación y la socialización, en lo que el marco del aula de clases se extienda al entorno, la familia y la comunidad en general. (13)

1.2 Software educativo. Medio de enseñanza aprendizaje.

El desarrollo acelerado de la ciencia y la tecnología en el campo de la informática ha provocado que en la actualidad se implementen aplicaciones informáticas en todos los sectores de la sociedad. En la educación se ha impulsado el uso de la tecnología educativa en función de potenciar la calidad del PEA, como la vía esencial para la formación de las nuevas generaciones, posibilitando la transmisión de conocimientos de una forma más amena, interactiva, integradora, diferenciada, reguladora y activa que el resto de los medios de enseñanza. Acerca de esta temáticas se han emitido diversos criterios que definen al software educativo como cualquier aplicación informática, sin advertir las condiciones que debe tener para su vinculación al PEA. (14)

Los software educativos (SE), se definen de forma genérica como aplicaciones o programas computacionales que facilitan el PEA. Algunos autores lo conceptualizan como cualquier programa computacional cuyas características estructurales y funcionales sirven de apoyo al proceso de enseñar, aprender y administrar, o el que está destinado a la enseñanza y el auto aprendizaje, y además permite el desarrollo de ciertas habilidades cognitivas. (15)

1.2.1 Características del software educativo.

El SE es concebido con un propósito específico: apoyar la labor del profesor en el proceso de aprendizaje de los estudiantes. Además de sus características computacionales, estas deben contener elementos metodológicos que orienten el proceso de aprendizaje. Son programas elaborados para ser empleados por computadores, generando ambientes interactivos que posibilitan la comunicación con el estudiante, la facilidad de uso es una condición básica para su empleo, debiendo ser mínimos los conocimientos informáticos para su utilización. Debe ser un agente de motivación para que el alumno pueda interesarse en este tipo de material educativo e involucrarlo. (16)

El SE individualiza el trabajo, se adapta al ritmo de trabajo de cada estudiante, y se pueden adaptar sus actividades según las actuaciones de los alumnos. (17)

1.3 Inteligencia artificial.

La IA es un campo de estudio que busca explicar y emular inteligencia, desarrollándola en términos de procesos computacionales que, si son utilizados correctamente por un programa, pueden exhibir un comportamiento inteligente. La IA pretende modelar a través de ordenadores la inteligencia del ser humano para resolver problemas de manera inteligente. Las personas

inferen y aprenden, y esto lo convierten en conocimiento. La IA imita estos aprendizajes del hombre creando modelos para lograr la solución de los problemas. (18)

El propósito de la IA no es la creación de agentes u hombres artificiales, como en la ciencia ficción. Tampoco intenta crear hombres o artefactos con sus mismas sensibilizaciones. Las propuestas sobre su desarrollo apuntan a la creación de modelos para soluciones inteligentes de problemas en dominios específicos. El propósito en realidad es la creación de sistemas inteligentes, empleando este concepto, no en toda su extensión, sino en lo referente a lo cognitivo. (19)

1.3.1 Técnicas de inteligencia artificial.

Uno de los más rápidos y sólidos resultados que surgieron en las investigaciones de la IA fue que la inteligencia necesita conocimiento. El conocimiento posee algunas propiedades poco deseable pues es voluminoso, es difícil caracterizarlo con exactitud, cambia constantemente, y se distingue de los datos en que se organiza de tal forma que se corresponde con la forma en que va a ser usado. Con los puntos anteriores se concluye que una técnica de IA es un método que utiliza conocimiento y debe ser comprendido por las personas que lo proporcionan. (20)

En muchos programas, los datos pueden adquirirse automáticamente en muchos dominios de la IA, la mayor parte del conocimiento que se suministra a los programas lo proporcionan personas, haciéndolo siempre en términos que ellos comprenden, puede modificarse fácilmente para corregir errores, y puede usarse en gran cantidad de situaciones aun cuando no sea totalmente preciso o completo. (20)

1.4 Sistemas tutoriales inteligentes.

Los autores Zulma Cataldi y Fernando J. Lage han hecho un resumen de los principales conceptos que se han dado sobre STI, algunos de ellos son:

“Sistema de software que utiliza técnicas de inteligencia artificial para representar el conocimiento e interactúa con los estudiantes para enseñárselo”.

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

“Sistema que incorpora técnicas Inteligencia Artificial a fin de crear un ambiente que considere los diversos estilos cognitivos de los alumnos que utilizan el programa”. (21)

Las definiciones de estos conceptos por varios autores permitieron darle seguimiento al objetivo propuesto, pues se desea elaborar un sistema que permita transmitir el conocimiento a

los alumnos mediante un proceso interactivo según sus características, teniendo en cuenta su estilo de aprendizaje, para una mejor explicación, la forma en que perciben la información y procesan los conocimientos. El sistema intentará simular la forma en que un profesor guiará al alumno una vez que conozca su forma de estudiar y aprender.

1.4.1 Arquitectura de los STI.

Las nuevas tecnologías de información y comunicación, ofrecen al ámbito educativo las condiciones necesarias para apoyar las labores de docencia e investigación, actualmente se hace más fácil construir y hacer uso de materiales de apoyo didáctico, desde libros electrónicos, sistemas tutoriales, hasta ambientes virtuales orientados a la investigación de los procesos educativos. Los STI se originaron en el momento que la IA estaba trabajando en el trascendente objetivo de imitar la inteligencia natural mediante la creación de máquinas que pensarán como los humanos. (22)

Siendo así, los STI representan un aporte concreto de la IA al campo educativo, también conocidos como sistemas de diálogo; son una herramienta didáctica que permite personalizar el PEA por medio de la aplicación de la experiencia y de la metodología pedagógica de un experto en un dominio específico. Se consideran "inteligentes" porque realizan acciones pedagógicas acerca de la forma de enseñar y mantienen información referente a las necesidades del estudiante. Un aspecto importante de estos sistemas es el de permitir desarrollar procesos mentales de índole superior tales como la resolución de problemas. (23)

Las características principales de un STI son: promover una respuesta activa en el alumno, informar al alumno sobre su desempeño, permitir un avance del aprendizaje de manera autónoma, y promover la eficiencia y eficacia del alumno en el trabajo. (24)

Los STI están compuestos por un módulo del dominio, un módulo del alumno y el módulo pedagógico, que operan de forma interactiva y se comunican a través de un módulo central que suele denominarse módulo entorno.

Módulo del Estudiante: El módulo del estudiante está presente en todos los trabajos en los que se describe la arquitectura básica de un STI, y representan características del estudiante tales como: el estilo de aprendizaje, el nivel de conocimiento, la información personal, o la combinación de algunas de ellas. (25)

Módulo del Dominio: El módulo del dominio, denominado también por muchos autores como módulo experto, proporciona los conocimientos del dominio. Satisface dos propósitos diferentes, en primer lugar, presentar la materia de la forma adecuada para que el alumno

adquiera las habilidades y conceptos, lo que incluye la capacidad de generar preguntas, explicaciones, respuestas y tareas para el alumno. En segundo lugar, el módulo del dominio debe ser capaz de resolver los problemas generados, corregir las soluciones presentadas y aceptar aquellas soluciones válidas que han sido obtenidas por medios distintos. En este módulo, el conocimiento a ser enseñado por el STI debe organizarse pedagógicamente para facilitar el PEA. (26)

Módulo Pedagógico: Decide qué, cómo y cuándo enseñar los contenidos almacenados en el STI, adaptando sus decisiones pedagógicas a las necesidades del estudiante. Algunos autores le denominan módulo tutor, ya que es el encargado de comparar las características de los estudiantes con el contenido a enseñar y elegir la mejor forma de tomar las decisiones pedagógicas oportunas, adaptándose en cada momento al estudiante. (27)

El **Módulo Entorno** gestiona la interacción de los otros componentes del sistema y controla la interfaz persona computadora. (28)

Luego del estudio realizado de la arquitectura definida para los STI se determinó que esta permite vincular el estudiante como actor principal en el PEA en un módulo del estudiante y al profesor como guía para la mejora en la educación en un módulo pedagógico.

1.4.2 Los sistemas de aprendizaje basado en casos. Su utilización en el desarrollo de tutoriales inteligentes.

El Razonamiento Basado en Casos (RBC) se basa en almacenar, recuperar y reutilizar las soluciones a problemas semejantes anteriormente resueltos. Las raíces del RBC hay que buscarlas en ciertos resultados de psicología donde se muestra que, en muchas ocasiones, los seres humanos resuelven problemas sobre la base de experiencias pasadas y no a partir de un conocimiento profundo del problema en cuestión. Los médicos, por ejemplo, buscan conjuntos de síntomas conocidos, los ingenieros toman muchas de sus ideas de soluciones previas ya construidas con éxito, o los programadores expertos reutilizan esquemas más o menos abstractos de las soluciones que conocen. (29)

El ciclo del RBC puede ser dividido en 4 procesos claramente diferenciados, dichos procesos son los siguientes:

Recordar los casos similares al que analizamos.

Reutilizar la información y el conocimiento que se tiene en este caso para resolver el problema.

Revisar la solución propuesta.

Guardar las partes de esta experiencia que puedan ser útiles para la resolución de futuros problemas. (30)

Ventajas y Desventajas de los sistemas basados en casos.

Ventajas

- ✓ Adquisición de conocimiento. La unidad básica del conocimiento es el caso. Los seres humanos por lo general articulan su conocimiento mediante ejemplos de problemas y soluciones anteriores (casos), más que por medio de reglas específicas y abstractas.
- ✓ Permite proponer soluciones a problemas rápidamente. Esto lo logra ya que las respuestas no se derivan a partir de cero, sino de casos resueltos previamente.
- ✓ Si la misma situación se presenta repetidamente, no se tiene que construir o generar la misma solución a partir de cero.
- ✓ Se centra en las características o partes más importantes del problema.

Desventajas

- ✓ Confía ciegamente en los casos previos almacenados en su memoria para intentar proponer su solución.
- ✓ Puede ser que no recupere el caso más apropiado para la solución del nuevo caso. (31)

1.5 Análisis de soluciones existentes.

El desarrollo de los STI, se caracteriza por la inclusión de experiencia adicional relacionada con el entorno de aprendizaje del estudiante, métodos y técnicas de enseñanza. Con estas nuevas características se han desarrollado sistemas más flexibles, adaptados a los intereses del estudiante y con métodos pedagógicos que facilitan el proceso de aprendizaje. A continuación se destacan algunos STI existentes a nivel nacional e internacional donde se evidencia el uso de la arquitectura definida anteriormente, aunque los sistemas son para el estudio y no para la resolución de ejercicios permitió identificar que se pueden adaptar a las características propias de un estudiante y dar inicio a la construcción de una nueva aplicación siguiendo la experiencia de los mencionados a continuación.

1.5.1 Soluciones existentes a nivel internacional.

CircSim: Es un STI que fue desarrollado en conjunto por el Departamento de Ciencias de la Computación del Instituto Tecnológico de Illinois y el Departamento de Fisiología del Colegio de Medicina. Este tutor es el más avanzado en su tipo, y se utiliza en el Colegio de Medicina para complementar las clases teóricas sobre problemas cardiovasculares. (32)

AGT (*Advanced Geometry Tutor*¹), es un STI para el uso en clases de geometría avanzada, desarrollado en la Universidad de Pittsburgh a través de la Fundación Nacional de Ciencias y el Centro de Investigaciones Interdisciplinarias en entornos de aprendizaje constructivo. Se basa en comparar dos estrategias de resolución de problemas, encadenamiento hacia adelante y encadenamiento hacia atrás, para determinar el aprendizaje de los estudiantes sobre teoremas de la geometría, con el fin de determinar qué estrategia acelera más el aprendizaje. (32)

1.5.2 Soluciones existentes a nivel nacional.

STIITS: Sistema Tutorial Inteligente para Diagnóstico y Tratamiento de Infecciones de Transmisión Sexual (ITS), que fue desarrollado por la Universidad de Cienfuegos, en conjunto con especialistas de segundo grado del hospital Dr. Gustavo Aldereguía Lima, que hicieron labor de expertos en la validación de la base de conocimientos para el posterior diagnóstico de las enfermedades, así como aportar todos los datos para la realización del tutorial. STIITS responde a una necesidad social de la provincia de Cienfuegos, y también nacional por sus posibilidades de empleo en la docencia. No existe algún otro software en el país con características similares para el mismo fin. (33)

VIRTEVALL. Metodología de evaluación para aprendizaje autónomo de idiomas extranjeros: este STI prevé crear un ambiente alternativo de evaluación para aprendizaje autónomo de idiomas extranjeros, que favorezca el trabajo de los estudiantes de manera independiente en sus debilidades, y profundizar en temas que sean de su interés en el aprendizaje del idioma inglés. Fue creado en el 2009 en la Universidad de las Ciencias Informáticas. (34)

HESEI: es una herramienta computacional para elaborar sistemas de enseñanza aprendizaje inteligente, la cual utiliza técnicas de IA y mapas conceptuales, con el objetivo de adaptar el sistema de enseñanza- aprendizaje, a través de una interfaz visual, a las características del alumno. Permite a expertos no especialistas en computación crear sistemas de enseñanza aprendizaje inteligentes que aborden contenidos de cualquier especialidad y nivel de enseñanza. (35)

1.5.3 Valoración de las herramientas.

Se han desarrollado STI en áreas como: Medicina, Teoría de Grafos, Análisis, Diseño de Sistemas y Estructura de Datos. Todos estos sistemas ayudan a mejorar el rendimiento,

¹ Tutor de Geometría Avanzada

convirtiéndose en sistemas muy eficaces para el aprendizaje, pero ninguno de ellos contribuye a la mejoría del PEA en la asignatura Ingeniería de Software.

La herramienta HESEI desarrollada en la Universidad de las Villas posibilita la creación de tutoriales inteligentes, pero no cuenta con el estilo de aprendizaje de la persona, está desarrollada en Delphi y no permite crear aplicaciones web, no permite resolver ejercicios, sino que es un medio para estudiar los contenidos de los materiales que deseen insertarse y no se adapta a la forma de aprender de los alumnos, es decir sus gustos. La herramienta cuenta con una base de conocimiento permitiendo al sistema escoger la forma más apropiada para enfrentar al estudiante, sin embargo hace más lento el sistema y la respuesta se tardaría buscando el caso más apropiado, de ahí que se requiera desarrollar un STI que permita a los estudiantes consolidar sus conocimientos y resolver las tareas docentes a partir de sus características centradas en el estilo de aprendizaje.

Fundamentación de las tecnologías, metodologías y herramientas.

1.6 Metodologías de desarrollo de software.

La metodología se define como un conjunto de métodos orientados para lograr una meta propuesta. Son los procesos que organizados en conjunto proporcionan una secuencia a seguir para la obtención del producto final (36). Las metodologías de desarrollo de software son un factor importante en la obtención de productos con calidad en el tiempo requerido. En la actualidad existen diversas metodologías de desarrollo, con características comunes pero también con significativas diferencias, estas se pueden clasificar en: (37)

Metodologías Robustas: Las metodologías robustas son aquellas que están guiadas por una fuerte planificación y control del proyecto durante todo el proceso de desarrollo; llamadas también metodologías clásicas o tradicionales, donde se realiza una especificación precisa de requisitos, modelado y una intensa etapa de análisis y diseño antes de la construcción del sistema. (38)

Estas metodologías son muy utilizadas por equipos de desarrollo con poca experiencia o que necesitan seguir una guía exhaustiva en el desarrollo de software, debido a todas las actividades que propone, los artefactos y la documentación que genera en el proceso de desarrollo de software en cada una de las etapas que presenta la metodología; además de la planificación y control que recomienda en el desarrollo de proyectos. De esta forma las metodologías robustas son capaces de guiar detalladamente a los equipos de proyecto en el desarrollo de sistemas informáticos. (39)

Metodologías Ágiles: Las metodologías ágiles forman parte del movimiento de desarrollo de software conocidas anteriormente como metodologías livianas, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil por la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

A continuación se realiza un estudio de las posibles metodologías a utilizar en el desarrollo del sistema a construir.

Programación Extrema (XP²):

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (40)

Entre las principales **características** que distinguen la metodología XP de otras se encuentran:

Iterativa: El proyecto se divide en versiones y estas a su vez en ciclos. En cada ciclo se desarrollan historias de usuarios que añaden funcionalidades al sistema. (41)

Compleja: XP propone un conjunto de técnicas (denominadas prácticas) para ser utilizadas durante el desarrollo. Cada técnica por separado aporta escaso valor al proyecto, algunas son complejas y de difícil implantación.

Cuando usar XP:

- ✓ Cuando los clientes no tienen ideas claras de los requisitos y los van cambiando.
- ✓ Para proyectos con poco personal (entre 2 y 10 programadores).
- ✓ Para proyectos de riesgo: fecha fija de entrega, algo nunca hecho por el grupo, algo nunca hecho por la comunidad de desarrolladores. (42)

La metodología XP se basa en cuatro fases fundamentales: planificación y exploración, diseño, implementación y pruebas.

² XP: Extreme Programming

Planificación del proyecto.

La planificación es la etapa inicial de proyecto en XP. En este punto se comienza a interactuar con el cliente y el resto del grupo de desarrollo para descubrir los requerimientos del sistema. En este punto se identifican el número de iteraciones al igual que se plantean ajustes necesarios a la metodología según las características del proyecto.

Las historias de usuario son utilizadas como herramienta para dar a conocer los requerimientos del sistema al equipo de desarrollo. Son pequeños textos en los que el cliente describe una actividad que realizará el sistema, la redacción se hace bajo la terminología del cliente, no del desarrollador, de forma que sea clara y sencilla, sin profundizar en detalles. En esta fase la creación de un plan de iteraciones permite dividir el desarrollo del sistema en etapas para facilitar su realización, por lo general constan de más de tres etapas, las cuales toman nombre de iteraciones, la duración ideal de una iteración es de una a tres semanas. (43)

Diseño

En esta fase se diseñan las historias de usuario que el cliente ha seleccionado para la iteración a realizar, se considera que no es posible tener un diseño completo del sistema sin errores desde el principio. El segundo motivo es que dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del sistema para luego modificarlo, se considera un desperdicio de tiempo.

Una de las partes más importantes de esta fase es lograr simplicidad en el diseño, pues se considera que un diseño sencillo se logra más rápido y se implementa en menos tiempo. La idea es realizar un diseño sencillo que cumpla con los requerimientos de las historias de usuario. En esta fase se elaboran las tarjetas de clase, responsabilidad y colaboración (CRC). (44)

Implementación

La fase de desarrollo o codificación es un proceso que se realiza en forma paralela con el diseño y está sujeto a varias observaciones consideradas por algunos expertos como la rotación de los programadores o la programación en parejas. En el desarrollo de las historias de usuarios, es necesaria la presencia del cliente, sin olvidar que son los clientes quienes crean las historias de usuarios y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará. La implementación debe hacerse bajo estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión. (45)

Pruebas

La fase de pruebas permite comprobar el funcionamiento de los códigos que se vayan implementando y así detectar fallas en el sistema. Permite crear las pruebas necesarias en el momento de encontrar un error antes de corregirlo, de esta forma el cliente logrará tener completamente claro cuál fue el error y dónde se encontraba el mismo, para solucionarlo en conjunto con el equipo de desarrollo y evitar volver a cometerlo. (46)

Proceso Unificado de Desarrollo

El Proceso Unificado de Desarrollo (RUP), es una metodología robusta que divide en 4 fases el desarrollo del software:

- ✓ Fase de Inicio, tiene como objetivo determinar la visión del proyecto.
- ✓ Fase de Elaboración, tiene como objetivo determinar la arquitectura óptima.
- ✓ Fase de Construcción, tiene como objetivo principal obtener la capacidad operacional inicial.
- ✓ Fase de Transición, cuyo objetivo es llegar a obtener la revisión del proyecto y pretende garantizar un producto preparado para la entrega a la comunidad de usuarios. (47)

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones y se reproducen en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Principales características:

- ✓ Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- ✓ Desarrollo iterativo y administración de requisitos.
- ✓ Uso de arquitectura basada en componentes, control de cambios y modelado visual del software.
- ✓ Verificación de la calidad del software. (48)

1.6.1 Selección de la metodología a utilizar.

Al analizar metodologías que se utilizan para el desarrollo de aplicaciones web, se selecciona XP para el desarrollo del módulo basado en PostgreSQL. El uso de la metodología ágil, se centra en los miembros del equipo y su interacción, en la entrega rápida de versiones de software funcional, en la colaboración constante del cliente y la facilidad para manejar los cambios, dándole menor importancia a las herramientas y la documentación. Esta metodología tiene como objetivo lograr la satisfacción del cliente y potenciar el trabajo en grupo. Es la

metodología que se ajusta al software educativo que se desea desarrollar, pues es un proyecto con un equipo de desarrollo pequeño y corto período de duración, donde el cliente forma parte del equipo y toma decisiones junto a éste. Presenta una programación organizada, donde el código será revisado continuamente mediante la programación en parejas, lo que hace posible que exista una menor tasa de errores. Propone además realizar pruebas, tanto unitarias, como de aceptación.

1.7 Lenguaje Unificado de modelado. UML 2.1.

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, permite visualizar, especificar, construir y documentar un sistema. Se puede aplicar en el desarrollo de software sin especificar que metodología o proceso usar. UML no es programación, solo se realizan diagramas para representar algunos requisitos y permitir a los creadores de sistemas generar diseños que capturen sus ideas y sea fácil de comprender por otras personas. En el sistema propuesto permitió conformar las tablas de la base de datos, el modelo de dominio, el diagrama de clases y el diagrama de despliegue en conjunto con la herramienta CASE seleccionada, posibilitando un mejor entendimiento de cómo quedaría la aplicación y una forma no funcional de su estructura.

1.8 Herramienta CASE.

Se han creado varias herramientas para el estudio de la Ingeniería de Software con el objetivo de desarrollar programas. Las herramientas CASE³ (Ingeniería de Software Asistida por Computadora) son utilizadas para la automatización del desarrollo de software, contribuyendo así a elevar la productividad y la calidad en el desarrollo de los sistemas informáticos. Dentro de las herramientas CASE orientadas a UML se encuentran: Visual Paradigm y Rational Rose.

1.8.1 Visual Paradigm. 6.4

Visual Paradigm (VP) es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis, diseño, implementación y pruebas. Permite construir diagramas de diversos tipos, generar código desde diagramas y generar documentación. Es importante señalar que la herramienta de modelado VP no es gratuita, pero la compañía Visual Paradigm UML Community tiene disponible distintas versiones, y facilita licencias especiales para fines académicos sin interés de lucro. (49)

Características principales:

³ *Computer Aided Software Engineering*

- ✓ Entorno de creación de diagramas para UML.
- ✓ Uso de un lenguaje estándar común en todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa (versión profesional) e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad en múltiples plataformas. (50)

1.8.2 Rational Rose.

Es una de las más poderosas herramientas CASE de modelado visual para el análisis y diseño de sistemas, se utiliza para modelar un sistema antes de proceder a construirlo y cubre todo el ciclo de vida de un proyecto.

Rational Rose tiene las características siguientes:

- ✓ Capacidad de análisis de calidad de código.
- ✓ Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requisitos de aplicación a través de diseños lógicos y físicos.
- ✓ Integración con otras herramientas de desarrollo de Rational.
- ✓ Publicación web y generación de informes para optimizar la comunicación dentro del equipo.

Desventajas:

- ✓ Necesidad de alta capacidad de procesamiento.
- ✓ Genera código en lenguaje Ada, ANSI C++, C++, CORBA, Java y Visual Basic, con funciones configurables de sincronización entre los modelos y el código, no así para el lenguaje PHP. (51)
- ✓ La herramienta Rational Rose no tiene soporte para la plataforma PHP. (52)

1.8.3 Selección de la herramienta CASE.

Tras el análisis de las herramientas mencionadas se selecciona la herramienta Visual Paradigm 6.4, pues permite modelar diagramas, generar código desde diagramas y generar documentación. La herramienta soporta múltiples usuarios trabajando sobre el mismo proyecto. Esta herramienta es multiplataforma ya que puede ser utilizada en cualquier sistema operativo y puede soportar el ciclo de vida completo de un software, permitiendo construir

aplicaciones de calidad a un menor costo.

1.9 Lenguajes de programación.

Java: Ofrece toda la funcionalidad de un lenguaje potente, se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje. Este elimina muchas de las características de otros lenguajes, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles, entre las que se encuentran mecanismos para garantizar la seguridad durante la ejecución de la aplicación comprobando, antes de ejecutar el código, contiene un gestor de seguridad que puede restringir el acceso a los recursos del sistema en desarrollo. Este lenguaje de programación es lento pues debe interpretar el código ante de ejecutar los métodos y gracias a la tecnología JIT⁴, este proceso se lleva a cabo una única vez. (53)

En la actualidad su uso es promovido para el desarrollo de aplicaciones empresariales del lado del servidor, especialmente a través del estándar J2EE⁵, así como en dispositivos móviles(a través del estándar J2ME⁶). (54)

Sin embargo el uso del lenguaje tiene sus limitantes, los programas hecho en Java no tienden a ser muy rápidos, considerablemente son más lentos y puede tomar más espacio en memoria que otros lenguajes compilados y la apariencia de las aplicaciones de interfaz gráfica de usuario son muy diferentes a las otras aplicaciones., al ser un lenguaje de programación aprenderlo lleva tiempo y especialmente para los no programadores. Para su uso se necesita tener instalada una máquina virtual y utiliza una gran cantidad de memoria. (55)

PHP: Es un lenguaje interpretado de alto nivel impregnado en páginas HTML y ejecutado en el servidor. PHP inició como una modificación a Perl escrita por Rasmus Lerdorf⁷a finales de 1994. (56)

PHP es un lenguaje de script incrustado dentro de HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas (57). El código de PHP es abierto por lo que cuenta con un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones del lenguaje para ampliar sus capacidades. (58)

⁴ Just-In-Time Compilation

⁵ Java 2 Enterprise Edition es una plataforma de programación

⁶ Java 2 Micro Edition es una colección de Interfaz de programación de aplicaciones en java

⁷ Rasmus Lerdorf creador de la primera versión del lenguaje de programación PHP

PHP funciona sobre prácticamente todas las plataformas imaginables y garantiza una alta velocidad de ejecución, además de una excelente estabilidad. El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable. El sistema debe poseer protecciones contra ataques y PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini. (59)

1.9.1 Selección del lenguaje de programación a utilizar.

Para el desarrollo del módulo se analizaron diferentes alternativas de desarrollo. Desde la utilización de un único lenguaje como PHP utilizando sus ventajas, hasta lenguajes que no dependieran en absoluto de la máquina de ejecución como puede ser JAVA, pero sin perder de vista que el objetivo fundamental es la obtención de un lenguaje que fuese multiplataforma, libre y de fácil uso para los programadores.

Analizando algunas de las características fundamentales de los lenguajes JAVA y PHP, se ha seleccionado PHP pues permite crear aplicaciones robustas, estables y con perfecta seguridad. Su utilización no necesita tener instalado una máquina virtual la cual crearía demoras en la computadora y ocuparía mucho espacio en memoria, no muestra el código PHP sino envía el resultado en HTML al navegador y con gran rapidez. PHP permite obtener información de la base de datos de una forma muy rápida y eficiente a través de consultas SQL⁸ que si devuelven, modifican una cierta información pueden hacer los cambios necesarios para resolver el objetivo propuesto. El lenguaje permite crear métodos para la entrada y validez de los caracteres en los campos de la aplicación, sin embargo utilizando JavaScript esto podría verse afectado si el usuario tiene desactivado su uso en el navegador. Por el tiempo considerable para el desarrollo de la aplicación es el ideal, siendo engorroso el estudio en profundidad de otro lenguaje de programación.

1.10 Sistemas Gestores de Base de Datos.

Un Sistema Gestor de Base de Datos (SGBD) se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar la definición de los datos, su mantenimiento, su control, su privacidad y su manipulación. (60)

MySQL: Es un sistema de gestión de bases de datos relacional escrito en C y C++, probado con un amplio rango de compiladores diferentes que funciona además en varios sistemas operativos. Las funciones SQL están implementadas usando una librería altamente

⁸ Lenguaje de Consulta Estructurado

optimizada y deben ser tan rápidas como sea posible, normalmente no hay reserva de memoria tras toda la inicialización para consultas. (61)

Otras características del sistema gestor de base de datos son:

- ✓ Aprovechamiento de la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's⁹ en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y contraseñas, manteniendo buen nivel de seguridad en los datos. (62)

PostgreSQL: Es un sistema de gestión de bases de datos objeto relacional (ORDBMS¹⁰) basado en el proyecto POSTGRES, de la universidad de Berkeley. Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. Soporta la gran mayoría de las transacciones SQL, control concurrente, teniendo a su disposición varios enlaces con lenguajes como C, C++, Java, Python, PHP y otros. (63)

PostgreSQL soporta distintos tipos de datos: datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, etc. También permite la creación de tipos propios. Incorpora una estructura de datos y funciones de diversa índole. Permite la declaración de funciones propias, así como la definición de disparadores. Soporta el uso de índices, reglas y vistas. Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. (64)

PostgreSQL es un gestor magnífico, que posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios web que posean alrededor de 500.000 peticiones por día. (65)

1.10.1 Selección del Sistema Gestor de Base de Datos.

Se ha seleccionado PostgreSQL 8.4 como gestor de base de datos a utilizar, pues constituye una herramienta muy potente que maneja grandes volúmenes de información, es software libre, multiplataforma y tiene todas las características de los SGBD modernos. Tiene una alta

⁹ Interfaz de programación de aplicaciones

¹⁰ Object relational database management system

integridad, robustez, y extensibilidad, también brinda gran seguridad y posee una alta escalabilidad, haciéndolo idóneo para el sistema que se quiere desarrollar.

Consideraciones parciales.

A modo de conclusión, luego de haber realizado un análisis detallado sobre el objeto de estudio y una búsqueda detallada de los sistemas tutoriales inteligentes existentes, se detectó que ninguno cumplía con los requisitos necesarios para desarrollar el sistema. Por lo anteriormente planteado se ratifica la necesidad de desarrollar un sistema tutorial inteligente que permita la personalización de las tareas docentes teniendo en cuenta las características de los estudiantes para la asignatura Ingeniería de Software 2 a través de la metodología, herramientas y lenguaje de programación propuesto, siendo las más apropiadas, pues son fáciles de usar y ofrecen los beneficios necesarios para el desarrollo de la aplicación.

Capítulo 2. Descripción del sistema.

2. Introducción.

Después de haber elegido las herramientas, metodologías, técnicas y haber analizado el estado del arte, están las condiciones creadas para realizar la propuesta de solución al problema existente. Para la implementación de la aplicación se siguieron los pasos definidos por la metodología XP en el proceso de desarrollo de software, comenzando con la descripción de las historias de usuarios, artefacto de gran importancia para la programación del sistema. El capítulo presenta una valoración de las principales características del sistema, el mismo está centrado en desarrollar la fase de planificación del proyecto, propia de la metodología en cuestión.

2.1 Descripción del sistema. Propuesta de solución.

Para cumplimentar el objetivo propuesto al inicio de este trabajo, los estudiantes se identificarán en el sistema y una vez validado que sus datos sean correctos podrán realizar la encuesta para determinar su estilo de aprendizaje, siendo este el paso principal para su inserción en la aplicación. El sistema tiene como objetivo que el estudiante domine las habilidades orientadas en los estudios independientes o tareas de la asignatura Ingeniería de Software 2, los cuales serán definidos mediante un tema en específico y orientados según su característica o forma de percibir los conocimientos.

El sistema brindará la facilidad de obtener la bibliografía necesaria para realizar las tareas, siendo así las conferencias, materiales didácticos, las clases prácticas y vínculos a páginas de orientación sobre el tema. En caso del estudiante no poder resolver una cierta tarea, la aplicación será capaz de asignarle materiales o medios según el estilo de aprendizaje hasta que logre vencer las habilidades. Las tareas una vez resueltas podrán ser adjuntadas para su posterior evaluación, permitirá incorporar la nota del estudiante una vez evaluada y conocer la impresión causada por la aplicación a través de una encuesta de satisfacción realizada.

2.1.1 Resultados esperados del sistema.

En el evento base de Universidad 2014 inaugurado en la UCI se presentó un conjunto de principios para el diseño del tutorial, a partir del cual se determinaron los requisitos del sistema. Algunos de ellos se presentan a continuación.

- ✓ El STI tiene como objetivo proveer una guía de estudio a cada estudiante, adaptada a su estilo de aprendizaje; este debe demostrar que ha desarrollado las habilidades de la asignatura, por medio de las tareas docentes insertadas en la aplicación.

- ✓ Para el acceso al sistema, como parte del registro en el mismo, los estudiantes deben realizar una encuesta ya predefinida, (es la que se utiliza para la caracterización de estudiantes de primer año) para determinar su estilo de aprendizaje. (Ver [Anexos 1](#)).
- ✓ Para que un estudiante pueda acceder a las funciones que le facilita el sistema para resolver las tareas debe haberse registrado previamente y para su registro debe haber completado la encuesta.
- ✓ Las tareas docentes primarias serán similares para todos los estudiantes, pero se diferenciarán por el ejemplo que se le ofrezca y esta estará en función de su(s) estilo(s) de aprendizaje.
- ✓ El sistema permitirá al profesor calificar las tareas, otorgando una nota a cada estudiante que desarrolla una tarea.
- ✓ El STI utilizará razonamiento basado en casos, para buscar los casos más similares almacenados en la base de datos; es decir, la información del estudiante o el estilo de aprendizaje guardado en el rasgo predictor. Una vez encontrada la información podrá aplicarse al usuario autenticado en el sistema, de no encontrarse el caso en la tabla de la base de datos se conformará uno nuevo que almacenará nuevos datos.
- ✓ Un caso es una clase de la base de datos del sistema, está conformado por rasgos predictores que reflejan el/los estilo(s) de aprendizaje del estudiante que se registre en el sistema al realizar la encuesta mencionada anteriormente, y un rasgo objetivo que comprende la guía de estudio que garantizó el desarrollo de las habilidades en ese estudiante, los medios y vías para facilitar la resolución de las tareas orientadas e insertadas en la aplicación.
- ✓ El sistema brindará además facilidades para el acceso a la bibliografía necesaria para realizar las tareas (conferencias, materiales didácticos y vínculos a páginas de orientación sobre el tema, además de la orientación de las tareas (7).

El sistema le brindará al profesor todos los privilegios para trabajar con la aplicación. Conocerá información de los estudiantes registrados en cuanto a la forma que tienen para aprender, visualizar los contenidos, obtener la información y sus formas de estudio, por medio de la encuesta para la obtención del estilo de aprendizaje. El sistema le posibilitará insertar tareas para controlar el avance de los estudiantes en cuanto al desempeño de las habilidades. Podrá evaluar las tareas una vez resueltas y adjuntadas por los estudiantes, conociendo de esta forma si han vencido el contenido y conocerá información referente al trabajo en una encuesta de satisfacción que le ayudará a darle respuesta paulatinamente. Con este previo conocimiento sabrá cómo enfrentar las diferencias individuales de cada estudiante en el desarrollo de las conferencias, y podrá encontrar medios y métodos para impartir las clases que motiven la participación de estos en la asignatura Ingeniería de Software 2.

El sistema evidencia el uso de la arquitectura definida en el capítulo anterior para un tutorial inteligente; el nombre y los apellidos, usuario, grupo y estilo de aprendizaje obtenido para cada alumno registrado, representa el módulo del estudiante. Las tareas insertadas por los profesores, la asignatura, los temas correspondientes a esa asignatura, las habilidades del tema y los medios para darle solución a dichas tareas teniendo en cuenta las características de los estudiantes representan el módulo de dominio.

El módulo pedagógico se encarga de facilitar los medios para resolver las tareas según el estilo de aprendizaje del estudiante y propiciar la mejor vía adaptándose a sus necesidades, tomando decisiones pedagógicas sabe qué, cómo y cuándo aplicar la información almacenada. El módulo entorno es la vista que tienen los estudiantes para interactuar con la aplicación.

2.2 Personas relacionadas con el sistema.

Las personas que interactúan con este sistema son aquellas interesadas en obtener un resultado del mismo. Teniendo en cuenta las características del sistema podrán acceder a él dos tipos diferentes de usuarios, todos deben autenticarse con el nombre de usuario y la contraseña para acceder al sistema.

Estudiantes: Una vez autenticados en el sistema podrán realizar el cuestionario para la obtención del estilo de aprendizaje permitiéndole realizar seguidamente las operaciones en el sistema.

Profesores: Pueden acceder a la aplicación una vez se hayan autenticado en el sistema, tienen todos los privilegios.

2.3 Descripción del dominio.

Un Modelo de Dominio es una representación visual estática del entorno real del proyecto. Puede utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis, como paso previo al diseño de un sistema, ya sea de software o de otro tipo (66). El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema, y ayudar a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común. (67)

Siguiendo la explicación anterior, para una mejor comprensión de la aplicación se tiene un STI conformado por casos, los cuales representan a los estudiantes, los casos se dividen en rasgos predictores y rasgos objetivos. Los rasgos predictores representan el módulo del estudiante y están basados en el estilo de aprendizaje, obtenido a través de un cuestionario una vez que el estudiante se haya autenticado en el sistema. Los rasgos objetivos representan el módulo de dominio y el módulo pedagógico y son las tareas insertadas en la aplicación por los profesores, las tareas tienen habilidades y pertenecen al tema de una asignatura, las habilidades poseen habilidades específicas y dispondrán de medios para resolverlas si el estudiante posee alguna dificultad. Una vez resuelta la tarea será

adjuntada en la aplicación y evaluada por los profesores. En la figura 3 se muestra el modelo de dominio, donde queda recogida la información.

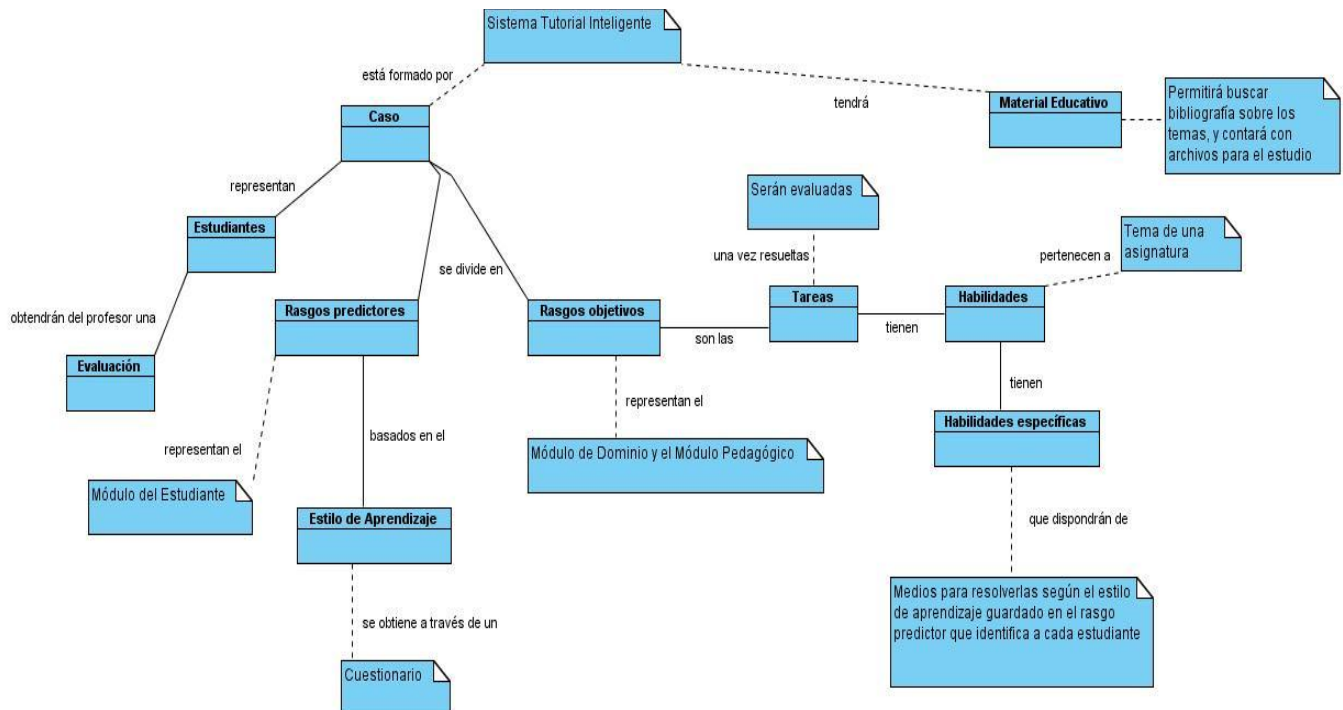


Figura 3. Modelo del dominio del sistema.

2.4 Lista de reserva del producto.

Según define la metodología XP, la Lista de Reserva del Producto (LRP) contiene los requisitos funcionales y los requisitos no funcionales del sistema.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir y los requisitos no funcionales son propiedades o cualidades que el producto debe tener, son las características que hacen al producto atractivo, usable, rápido y/o confiable. (68)

De acuerdo a estas definiciones de requisitos, se mencionan a continuación los requisitos que debe cumplir el sistema.

2.4.1 Requisitos funcionales.

RF 1: Autenticar usuario.

RF 1.1: Validar datos introducidos.

RF 2: Realizar encuesta de estilo aprendizaje.

RF 3: Registrar estudiantes.

RF 4: Ver listado estudiantes.

RF 5: Gestionar tarea de estudiantes.

RF 6: Asignar materiales para resolver tareas.

RF 7: Ver tareas.

RF 8: Subir tareas resueltas.

RF 9: Evaluar tareas.

RF 10: Mostrar evaluación de tareas.

RF 11: Gestionar material educativo.

RF 12: Ver material educativo.

RF 13: Realizar encuesta de satisfacción.

RF 14: Mostrar resultado encuesta de satisfacción.

RF 15: Gestionar asignatura.

RF 16: Gestionar tema.

RF 17: Gestionar habilidades.

RF 18: Gestionar habilidades específicas.

RF 19: Gestionar material para las habilidades específicas.

RF 20: Relacionar habilidad tarea.

RF 21: Mostrar tareas resueltas adjuntadas.

RF 22: Eliminar estudiante.

Requisitos funcionales para el razonamiento basado en casos.

RF 23: Insertar rasgo predictor.

RF 24: Buscar casos similares.

RF 25: Agregar caso en la base datos.

2.4.2 Requisitos no funcionales.

Software

Requisitos mínimos para el cliente: Debe tener instalado navegador web para acceder a la aplicación.

Requisitos mínimos para el servidor: Se requiere del gestor de base de datos PostgreSQL 8.4, servidor web Apache 2.2.12, PHP 5.3.0, pues el sistema se desarrolló sobre estas versiones, existiendo la posibilidad de incompatibilidades con versiones anteriores.

Hardware

Los requisitos mínimos para la ejecución de la aplicación se resumen en: procesador a 2.0 GHz de velocidad o superior, 512 MB de RAM o superior y 5 GB de espacio libre en disco duro.

Restricciones en el Diseño y la Implementación.

El diseño de la aplicación será basado en la metodología de desarrollo XP, con el uso del lenguaje de modelado UML.

Se usará como herramienta CASE Visual Paradigm para el modelado de los artefactos que se generan en las fases de la metodología seleccionada.

Se usará como lenguaje de programación PHP.

Se usará como sistema gestor de base de datos PostgreSQL.

Apariencia o interfaz externa

Las páginas de la aplicación no tendrán muchas imágenes y poseerán pocos colores.

Las páginas principales tendrán información que servirá de guía al usuario para trabajar con la aplicación una vez realice una operación.

El logo estará ubicado en la parte superior izquierda del sitio, exponiendo sus siglas y el significado en la parte derecha de la página.

Usabilidad

Las personas que interactuarán con el software serán estudiantes y profesores de la UCI.

La interfaz de la aplicación debe ser sencilla, facilitando al usuario que encuentre fácilmente lo que desea realizar, y con el objetivo de mantener todo el tiempo orientado al usuario debe brindar la posibilidad de mensajes, tanto para facilitar el trabajo con el sistema como en el caso de ocurrir algún error.

Seguridad

El acceso o cualquier manipulación del sistema, debe estar sometido a un proceso de autenticación del usuario donde será especificado el usuario y la contraseña.

El usuario y la contraseña serán validados mediante el servicio (Protocolo Ligero de Acceso a Directorios) LDAP¹¹ para comprobar que pertenecen al dominio uci.

Cada usuario en dependencia de ser profesor o estudiante tendrá asignado responsabilidades en el sistema, teniendo niveles de acceso al software.

2.5 Planificación del proyecto.

En esta fase se define el alcance general del trabajo. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán para su desarrollo, de igual forma se prueba la tecnología, y se exploran las posibilidades de la arquitectura del sistema. Toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

En esta fase, el cliente define lo que necesita mediante la redacción de sencillas historias de usuarios. Los programadores estiman los tiempos de desarrollo en base a esta información, y debe quedar claro que las estimaciones realizadas en esta fase son primarias, ya que estarán basadas en datos de muy alto nivel, y podrían variar cuando se analicen más en detalle en cada iteración. (69)

2.5.1 Historias de usuarios.

La especificación de las Historias de Usuario (HU) es la técnica utilizada para describir los requisitos del software mencionados anteriormente. Se trata de tarjetas en las cuales se detallan las características que deben cumplir cada una de las funcionalidades para su implementación, y así obtener una completa aceptación por parte de la Msc¹². Dayana Caridad Tejera Hernández en este caso el cliente.

A medida que se va dando cumplimiento a cada HU se presenta el software debidamente probado al cliente, para realizar las pruebas de aceptación de la funcionalidad implementada. Cuando se reúnen un número suficiente de funcionalidades que representan una versión útil y parcialmente completa de la aplicación se produce una liberación, lo cual es una versión funcional de la aplicación que aporta valor al negocio y que debe ser mantenida a la par que se desarrollan las siguientes funcionalidades. (Ver [Anexos 2.](#))

¹¹ Lightweight Directory Access Protocol

¹² Máster en ciencias

Plantilla de las historias de usuarios.

Historia de Usuario	
Número: Número de la HU	Nombre: El nombre de la HU, sirve para identificarla fácilmente entre los desarrolladores y los clientes.
Usuario: El usuario del sistema que utiliza o protagoniza la historia	
Prioridad en Negocio: Que tan importante es para el cliente.	Nivel de Complejidad: Que tan difícil es para el desarrollador.
Iteración Asignada: La iteración (liberación en nuestro proceso) a la que corresponde.	
Descripción: La descripción de la historia, detallando las operaciones del usuario y opcionalmente las respuestas del sistema.	
Observaciones: Algunas observaciones de interés, como glosario, información sobre usuarios, etc.	
Prototipo de Interfaz: Imagen de cada una de las interfaces relacionadas con la HU.	

2.6 Plan de iteraciones.

Después de ser escritas e identificadas las historias de usuario, el siguiente paso es especificar cuáles historias de usuario serán implementadas por cada iteración del sistema.

Iteración 1.

En esta iteración se implementarán las historias de usuario que por la importancia que tienen para el cliente, tienen prioridad con respecto a las otras. Al finalizar esta iteración se contará con las funcionalidades descritas en las historias de usuario 1, 2, 3, 4, 15, 16, 17, 18 y 22 referentes a la autenticación en el sistema, realizar encuesta para obtener el estilo de aprendizaje, el registro de los estudiantes, gestionar asignatura, gestionar tema de las asignaturas, gestionar habilidades, gestionar habilidades específicas y eliminar estudiantes del sistema. Esta versión del sistema tiene como objetivo mostrarle al cliente cómo va quedando la aplicación, para comprobar el grado de aceptación que tiene el producto.

Iteración 2.

El objetivo de esta iteración es la implementación de las funcionalidades de prioridad alta, media y baja que no fueron tratadas en la primera iteración. Al término de esta, se tendrán implementadas las

funcionalidades reflejadas en las historias de usuario 5, 7, 11, 12, 19, y 20 referentes a la gestión de las tareas de los estudiantes, ver listado de tareas, gestionar material educativo, gestionar material para las habilidades específicas y relacionar habilidad con una tarea. Al igual que en la primera iteración, al término de esta se contará con una versión de prueba del producto, versión número 2 que tendrá un objetivo similar a la anterior iteración.

Iteración 3.

En esta iteración se implementarán las funcionalidades restantes, con la culminación de la misma estarán desarrolladas las peticiones del cliente descritas en las historias de usuario 6, 8, 9, 10, 13, 14 y 21 referentes a la asignación de materiales para resolver las tareas, subir las tareas resueltas, evaluar las tareas, mostrar evaluación de las tareas, realizar encuesta de satisfacción, mostrar resultado de la encuesta de satisfacción y mostrar las tareas adjuntadas. Al término de esta iteración se contará con la versión 1.0 del producto final.

Plan de duración de las Iteraciones.

Iteraciones	Orden de las Historias de Usuario a implementar	Cantidad de tiempo de trabajo
1	Autenticar usuario.	2.2 semanas
	Realizar encuesta de estilo aprendizaje.	
	Registrar estudiantes.	
	Ver listado estudiantes.	
	Gestionar asignatura.	
	Gestionar tema.	
	Gestionar habilidades.	
	Gestionar habilidades específicas.	
Eliminar estudiante.		
2	Gestionar tarea de estudiantes.	2.5 semanas
	Ver tareas.	
	Gestionar material educativo.	
	Ver material educativo.	
	Gestionar material para habilidades específicas.	

	Relacionar habilidad tarea.	
3	Asignar materiales para resolver tareas.	2 semanas
	Subir tareas resueltas.	
	Evaluar tareas.	
	Mostrar evaluación de tareas.	
	Realizar encuesta de satisfacción.	
	Mostrar resultado encuesta de satisfacción.	
	Mostrar tareas resueltas adjuntas.	
Total de semanas		7

2.7 Plan de entrega.

Para facilitar la elaboración del plan de entrega para la fase de implementación, se acoplaron las funcionalidades referenciadas en cada una de las iteraciones en forma de módulos. Estos módulos facilitarán el trabajo del programador, ya que contribuyen a la organización del trabajo y a evitar la repetición innecesaria de código.

Composición de módulos.

Módulos	Historias de usuario que agrupa
1	Autenticar usuario.
	Realizar encuesta de estilo aprendizaje.
	Registrar estudiantes.
	Ver listado estudiantes.
	Gestionar asignatura.
	Gestionar tema.
	Gestionar habilidades.
	Gestionar habilidades específicas.
	Eliminar estudiante.
2	Gestionar tarea de estudiantes.
	Ver tareas.
	Gestionar material educativo.

	Ver material educativo.
	Gestionar material para habilidades específicas.
	Relacionar habilidad tarea.
3	Asignar materiales para resolver tareas.
	Subir tareas resueltas.
	Evaluar tareas.
	Mostrar evaluación de tareas.
	Realizar encuesta de satisfacción.
	Mostrar resultado encuesta de satisfacción.
	Mostrar tareas resueltas adjuntas.

Consideraciones parciales.

A partir del resultado del análisis de las encuestas realizadas se pudieron identificar las necesidades existentes para resolver los problemas en la asignatura Ingeniería de Software 2. Una vez obtenidas estas necesidades se describió como debería quedar el sistema propuesto y se identificaron las funcionalidades y cualidades con las que debe contar el sistema. Se definieron 25 requisitos funcionales los cuales fueron descritos en las historias de usuarios, permitiendo dar solución al sistema propuesto.

Capítulo 3. Diseño, Implementación y Pruebas.

3. Introducción.

En el presente capítulo se abordan las fases diseño, implementación y pruebas de la Metodología XP. Esta metodología plantea que la implementación debe realizarse de forma iterativa e incremental, permitiendo así que al final de cada iteración surja un producto funcional que debe ser probado y mostrado al cliente, permitiendo de esta forma lograr una constante retroalimentación desarrollador-cliente.

Uno de los artefactos fundamentales es la creación de las tarjetas CRC las cuales permiten brindar un mayor enfoque orientado a objetos. Además, este capítulo tiene dentro de sus objetivos, detallar las tres iteraciones llevadas a cabo durante la etapa de construcción del sistema, así como el diseño del mismo.

3.1 Diseño del sistema.

La Metodología de Desarrollo XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo (70). XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

Para el diseño de las aplicaciones, XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC siempre que sean útiles, tributen a la comprensión y no requieran mucho tiempo en su creación. No obstante el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación entre el equipo de desarrollo, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante.

XP establece prácticas especializadas que inciden directamente en la realización del diseño para lograr un sistema robusto y reutilizable tratando de mantener su simplicidad, es decir, crear un diseño evolutivo que vaya mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente. (71)

3.2 Tarjetas CRC. Cargo o Clase, Responsabilidad y Colaboración.

Las tarjetas CRC son utilizadas para representar las responsabilidades de las clases y sus interacciones (72). Se definen las tarjetas CRC con la finalidad de obtener un diseño simple y no incurrir

en la implementación de características que no son necesarias. En la figura 4 se muestra el diagrama de clases.

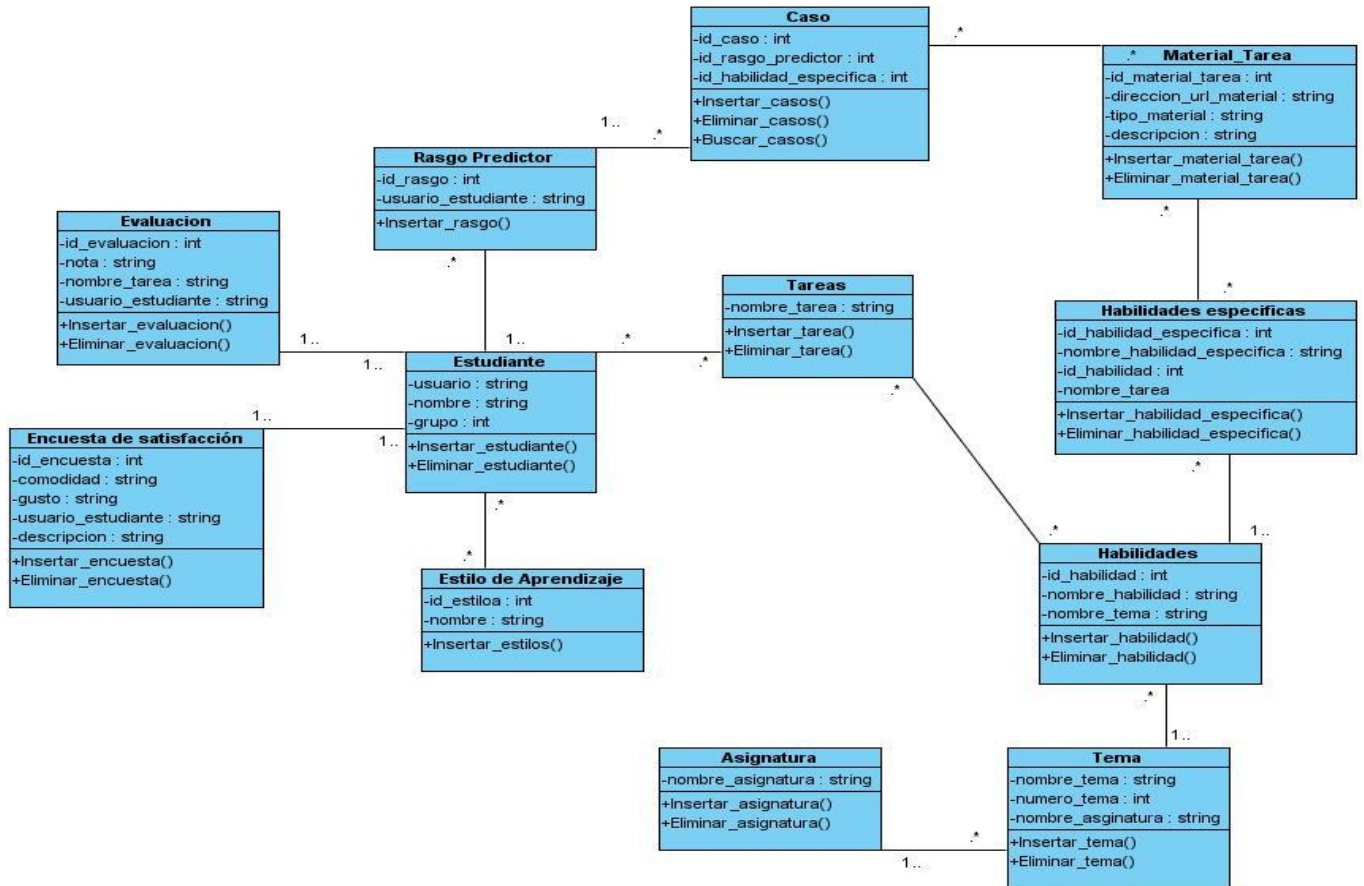


Figura 4. Diagrama de clases del sistema.

Plantilla de la Tarjeta CRC

Tarjeta CRC	
Nombre de la clase:	
Responsabilidad:	Colaboración:

Tabla 1. Tarjeta CRC clase estudiante.

Tarjeta CRC
Nombre de la clase: estudiante

<p>Responsabilidad:</p> <p>El estudiante se debe registrar en la base de datos guardándose su nombre, apellidos, usuario y estilo de aprendizaje después de realizar la encuesta en el inicio del registro</p>	<p>Colaboración:</p> <p>login, encuesta, cuestionarioA, cuestionarioB, cuestionarioC, cuestionarioD, cuestionarioE, cuestionarioF, cuestionarioG, cuestionarioH</p>
---	--

Tabla 2. Tarjeta CRC clase evaluación.

Tarjeta CRC	
Nombre de la clase: evaluación	
<p>Responsabilidad:</p> <p>La evaluación se realizará cuando un estudiante seleccione resolver una determinada tarea y la adjunte en el sistema, recogiendo toda la información</p>	<p>Colaboración:</p> <p>controladorEvaluarTarea, modelEvaluarTarea, evaluar, verEvaluacionP, vistaEvaluacionResult</p>

Tabla 3. Tarjeta CRC clase encuesta satisfacción.

Tarjeta CRC	
Nombre de la clase: encuesta_satisfacción	
<p>Responsabilidad:</p> <p>La encuesta de satisfacción permitirá almacenar la información de los estudiantes, en cuanto al trabajo con el sitio y su satisfacción con el mismo para resolver las tareas</p>	<p>Colaboración:</p> <p>esatisfaccion, controladorEncuesta, verEncuesta, vistaEncuesta</p>

Tabla 4. Tarjeta CRC clase tarea.

Tarjeta CRC	
Nombre de la clase: tarea	

<p>Responsabilidad:</p> <p>Las tareas podrán adicionarse en el sistema, eliminarse del sistema y ser vistas por los estudiantes y profesores. Los estudiantes podrán resolver las tareas</p>	<p>Colaboración:</p> <p>adicionarTareas, eliminarTareas, verTareas, tareas</p>
---	---

Tabla 5. Tarjeta CRC clase material tarea.

Tarjeta CRC	
<p>Nombre de la clase: material_tarea</p>	
<p>Responsabilidad:</p> <p>Los estudiantes para la resolución de las tareas podrán consultar materiales específicos por cada habilidad específica llegando a la resolución de los ejercicios</p>	<p>Colaboración:</p> <p>adicionarMaterialEspecifico, eliminarMaterialEspecifico, controladorSTI</p>

Tabla 6. Tarjeta CRC clase asignatura.

Tarjeta CRC	
<p>Nombre de la clase: asignatura</p>	
<p>Responsabilidad:</p> <p>La asignatura podrá adicionarse y eliminarse del sistema</p>	<p>Colaboración:</p> <p>adicionarAsignatura, eliminarAsignatura</p>

Tabla 7. Tarjeta CRC clase tema.

Tarjeta CRC	
<p>Nombre de la clase: tema</p>	
<p>Responsabilidad:</p> <p>El tema podrá adicionarse y eliminarse del sistema</p>	<p>Colaboración:</p> <p>adicionarTema, eliminarTema</p>

Tabla 8. Tarjeta CRC clase habilidad.

Tarjeta CRC	
Nombre de la clase: habilidad	
Responsabilidad: La habilidad podrá adicionarse y eliminarse del sistema y podrá relacionarse con una tarea de un tema en específico	Colaboración: adicionarHabilidad, eliminarHabilidad, relacionar

Tabla 9. Tarjeta CRC clase habilidad específica.

Tarjeta CRC	
Nombre de la clase: habilidad_específica	
Responsabilidad: La habilidad específica podrá adicionarse y eliminarse del sistema	Colaboración: adicionarHabilidadEsp, eliminarHabilidadEsp

Tabla 10. Tarjeta CRC clase material educativo.

Tarjeta CRC	
Nombre de la clase: material_educativo.	
Responsabilidad: Los materiales educativos podrán adicionarse, eliminarse del sistema y podrán ser observados por los estudiantes y profesores	Colaboración: adicionarMaterial, eliminarMaterial, materiales

3.3 Patrón de arquitectura y diseño.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño, para que una solución sea considerada un patrón debe poseer ciertas características.

Debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores.

Debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. (73)

La arquitectura del software es el diseño de más alto nivel de la estructura de un sistema. Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software. La arquitectura de software establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos del sistema. (74)

Uno de los factores que determina el éxito o fracaso de un sistema de software, es su arquitectura. Con esto se refiere a la estructura del sistema, las cuales están compuestas de elementos de software, propiedades externas visibles de estos elementos y las relaciones entre sí. El seleccionar debidamente una arquitectura de software, garantiza que el sistema de software cumpla con uno o varios atributos de calidad, que sea fácil de usar, confiable o seguro. Sin embargo, si la arquitectura no se diseña de forma apropiada, el sistema de software resultante no logrará sus objetivos. De nada sirve un sistema de software que no cumple con los tiempos de respuesta requeridos por el cliente, o que es complejo de modificar, difícil de usar o vulnerable a ataques.

Regularmente, no se conoce hasta el final del desarrollo del sistema de software, si éste cumplió o no con los atributos de calidad que se especificaron en los requisitos no funcionales. Dicho conocimiento tardío, implica tomar demasiados riesgos innecesarios, un ejemplo es, descubrir fallas en el sistema de software debido a que en la fase de diseño no se eligió apropiadamente una arquitectura. Para reducir tales riesgos y, como una buena práctica de ingeniería se seleccionó el patrón de arquitectura y diseño Modelo Vista Controlador (MVC). (75)

MVC: es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. (76)

El patrón de llamada y retorno MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el sistema de gestión de base de datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. (77)

3.3.1 Descripción del patrón MVC.

Modelo: Es la representación específica de la información con la cual el sistema opera. Representa las estructuras de datos y típicamente el modelo de clases contendrá funciones para consultar, insertar y actualizar información de la base de datos.

Vista: Presenta el modelo en un formato adecuado, usualmente la interfaz de usuario. Es la parte que utilizan los usuarios para interactuar con la aplicación.

Controlador: Responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo, actúa como intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para generar una página.

A partir del diagrama de paquetes mostrado en las figuras 5, 6 y 7 respectivamente, se describe la arquitectura del sistema formada por las capas: Modelo, Vista y Controlador, con los componentes que conforman cada capa evidenciándose así el uso del patrón seleccionado.

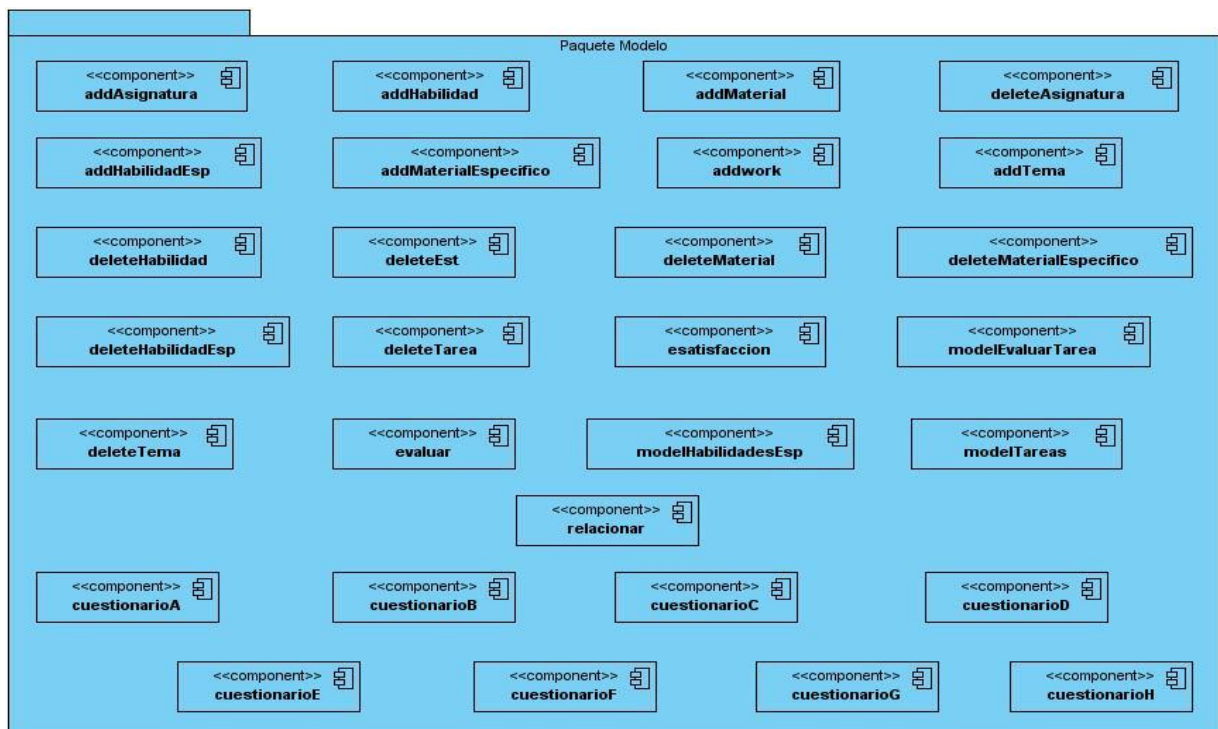


Figura 5. Componente modelo del patrón MVC.

Componente Modelo: es donde residen los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de la información.

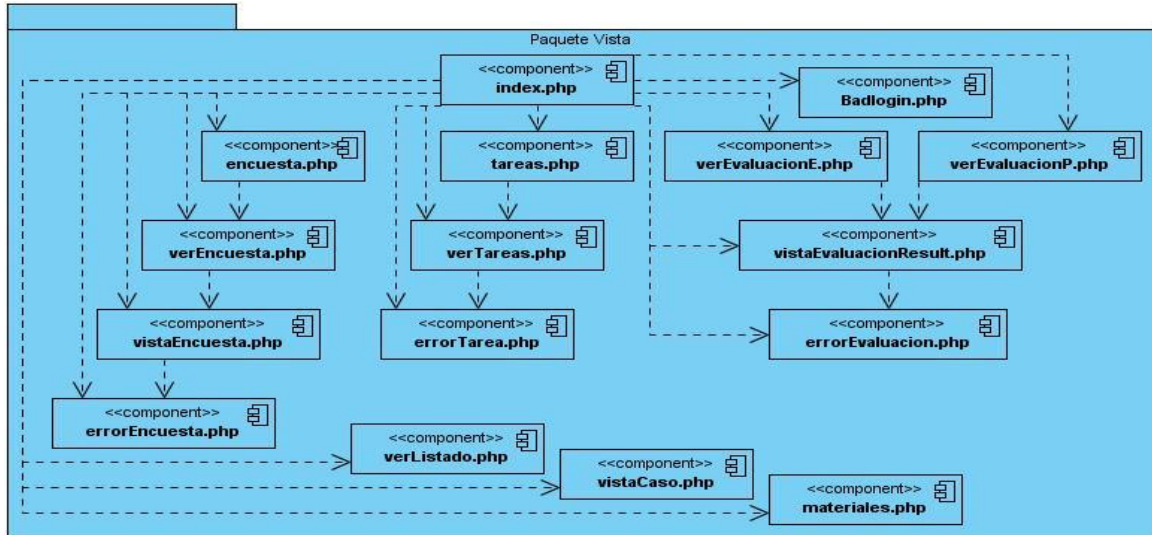


Figura 6. Componente Vista del patrón MVC.

Componente Vista: es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso.

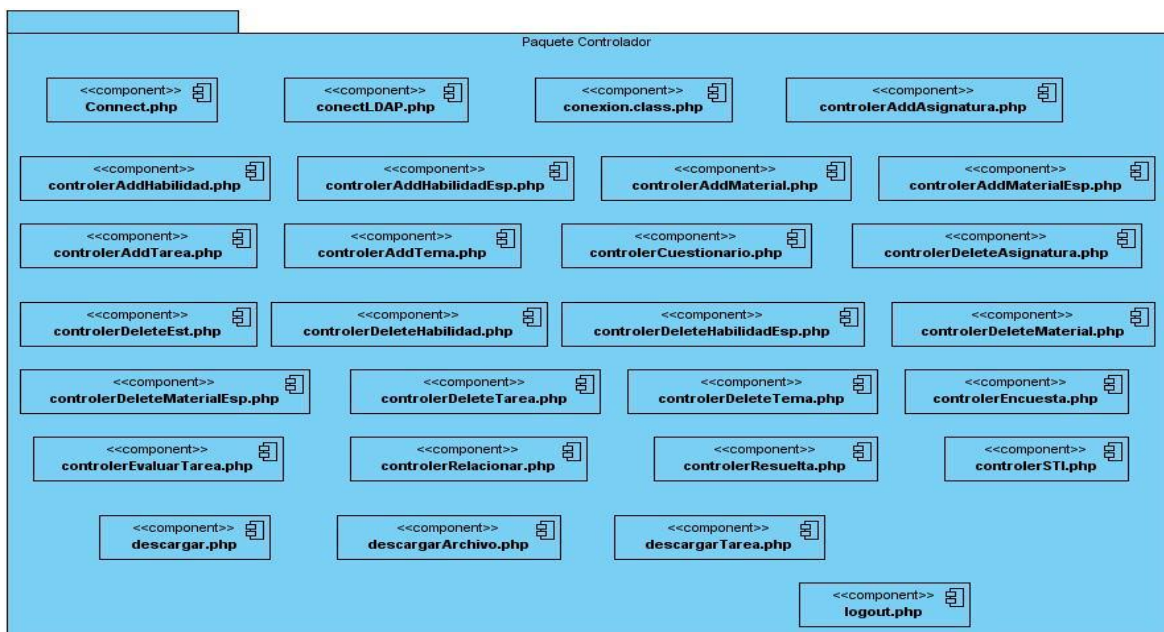


Figura 7. Componente Controlador del patrón MVC.

Componente Controlador: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Es donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos el almacenamiento o la recuperación de su información.

Servicios utilizados

LDAP es un conjunto de protocolos abiertos usados para acceder a información guardada centralmente a través de la red. LDAP organiza la información en un modo jerárquico usando directorios, los cuales pueden almacenar una gran variedad de información. (78)

LDAP

Comprobar usuario: este servicio realiza una búsqueda interna para comprobar si el usuario recibido por parámetro es válido en el directorio activo de la universidad.

Obtener datos del usuario: este servicio permite obtener todos los datos de un usuario determinado, entre ellos: el nombre y los apellidos.

3.4 Modelo de datos.

Un modelo de datos es una representación abstracta de los datos de un sistema u organización y las relaciones existentes entre ellos, más aún se puede decir que en cierta medida un modelo de datos describe la estructura de un sistema y tiene como propósito representar los datos de una manera comprensible. Es una propiedad estática que define entidades u objetos, propiedades o atributos de esas entidades y relaciones entre esas entidades. (79)

El modelo de datos se confeccionó en la herramienta VP, con la información obtenida en entrevista con la tutora, la aplicación sería parte de su doctorado por ello era necesario guardar la información con la que iba a trabajar el sistema y elaborar el modelo, el diseño en la herramienta CASE permitió generar las tablas de la base de datos. En la figura 8 se muestra como quedó conformado este modelo.

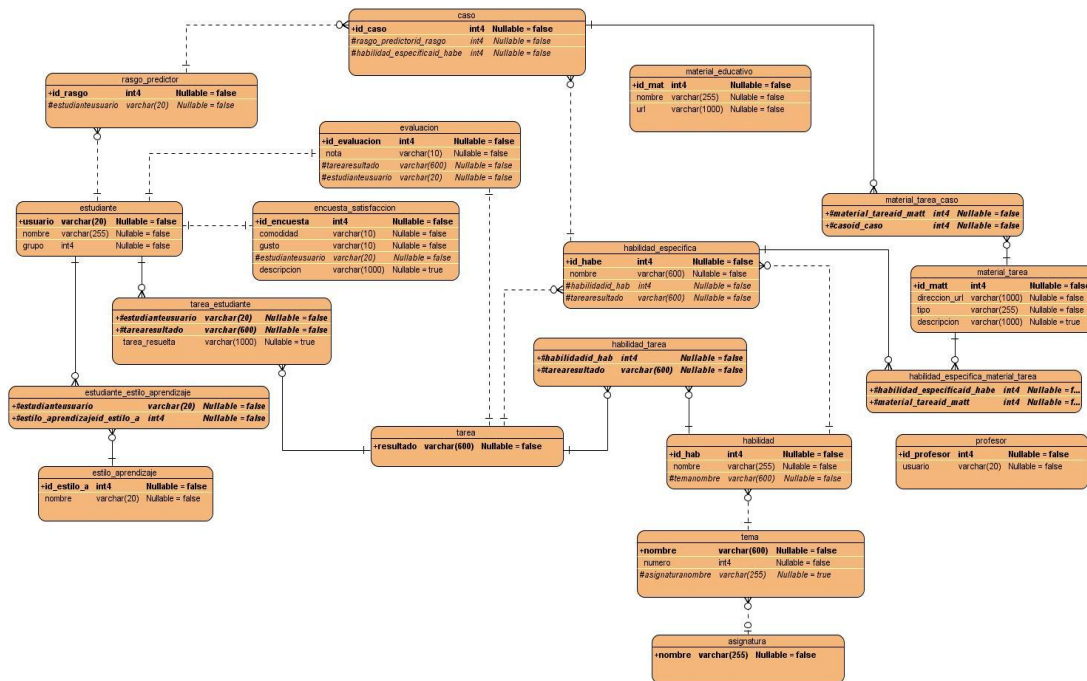


Figura 8. Modelo de datos del sistema.

3.5 Implementación del sistema.

Desarrollo de las iteraciones.

En la fase de Planificación se detallaron las HU correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las necesidades requeridas por el cliente. Durante el transcurso de las iteraciones se llevaron a cabo una revisión del plan de iteraciones y se modificaron en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de programación o ingeniería, asignando a un equipo de desarrollo (o una persona), responsable de su implementación, aplicando la práctica de la programación en parejas. Estas tareas no tienen que necesariamente ser entendidas por el cliente, pueden ser escritas en lenguaje técnico y son para el uso estricto de los programadores.

Teniendo en cuenta la planificación realizada con anterioridad, se llevó a cabo el desarrollo del sistema o fase de implementación según la metodología XP en tres iteraciones, obteniéndose como finalidad un producto con todas las restricciones y características deseadas por el cliente.

En la implementación, la utilización de la IA permitió utilizar los datos almacenados para conformar los casos, los cuales podían ser reutilizados si la información contenida en este, en cuanto a estilo de aprendizaje era igual a la información del estudiante que fuese a resolver alguna tarea y entraba por primera vez a la aplicación. La IA tiene como objetivo mostrar de una forma inteligente la información contenida en una base de datos. En el sistema se comporta de esta forma funcionando como un módulo pedagógico ya que se conocen las características del estudiante, y permite impartir los medios

como si fuera una persona, pero sin tratar de sustituir al ser humano. Se comporta como un tutor inteligente que se adapta a la forma que tiene el alumno para entender los conocimientos.

3.5.1 Tareas de la implementación.

Para la implementación del sistema se llevan a cabo una serie de tareas que no se encuentran comprendidas en las HU y que se han definido como tareas generales a realizar. Las tareas de la implementación (tareas de ingeniería) son muy importantes para el programador porque guían el proceso de desarrollo del sistema. (Ver [Anexos 3.](#))

3.6 Diagrama de despliegue.

El diagrama de despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Describen la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red y conexiones TCP/IP. (80)

Siguiendo la explicación anterior el sistema tutorial inteligente estará distribuido de la siguiente manera: una PC cliente donde el usuario hará uso de las distintas funcionalidades ofrecidas, accediendo al servidor de aplicaciones donde se encontrará instalado el sistema, mediante el protocolo HTTP y este se comunicará mediante la familia de protocolos TCP/IP con el servidor de base de datos donde se encontrará la base de datos de la aplicación, haciendo uso del servicio LDAP para obtener información de los usuarios. En la figura 9 se muestra el diagrama de despliegue del sistema.

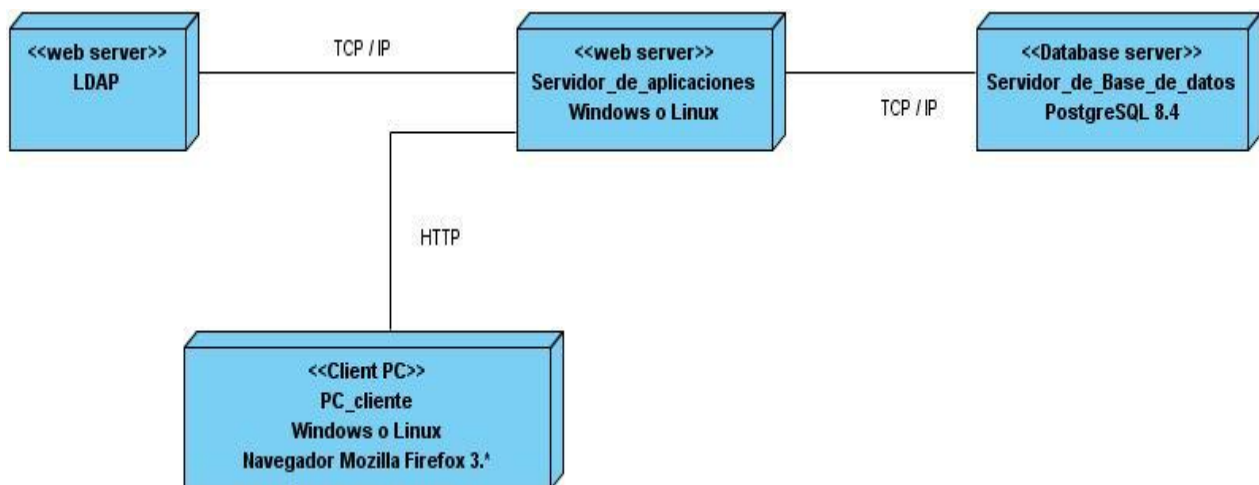


Figura 9. Diagrama de despliegue del sistema.

3.7 Pruebas.

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente, todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones. La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

3.7.1 Pruebas de aceptación.

Las Pruebas de Aceptación (PA) son pruebas de caja negra que se realizan a partir de las HU. Durante las iteraciones las historias de usuarios escogidas serán traducidas a prueba de aceptación. En ella se especifican la perspectiva del cliente y los escenarios para probar que la historia de usuario ha sido implementada correctamente. Una historia de usuario puede tener todas las pruebas de aceptación que desee para asegurar su funcionamiento. El objetivo específico de esta prueba es garantizar que los requisitos han sido cumplidos y que el sistema ha sido aceptable. (81) (Ver [Anexos 4.](#))

3.7.2 Resultados de las pruebas de aceptación.

En el desarrollo de las PA del software se comprobó el buen funcionamiento de las operaciones de la aplicación, sin embargo se detectó en una primera iteración que las palabras con tildes no eran mostradas por el sistema ni insertadas en la base de datos, la encuesta de obtención del estilo de aprendizaje regresaba a la sección de inicio en caso de dejar alguna pregunta en blanco y los campos de entrada de datos no estaban validados correctamente. Estas deficiencias fueron tomadas como no conformidades del usuario procediéndose a la subsanación de dichos errores.

3.8 Pruebas realizadas a los estudiantes.

Las pruebas realizadas en el epígrafe anterior permitieron comprobar el correcto funcionamiento de los requisitos desarrollados, aunque se detectaron ciertas no conformidades las cuales fueron corregidas en el momento de las pruebas, ventaja que permite la metodología XP. Se desea desarrollar una prueba a una pequeña muestra de estudiantes para dar respuesta a la última tarea de la investigación y comprobar el comportamiento de la aplicación, verificando que esté en condiciones de desarrollar las tareas que el usuario que lo adquiera planea llevar a cabo, en este caso los estudiantes y confirmando cumpla con los resultados detallados en el capítulo 2.

3.8.1 Verificación de la consistencia de los datos.

En la prueba de verificación se insertó un número finito de datos en el sistema completando las tablas de la base de datos, buscando que se ejecutara el ciclo completo de la solución, que incluye desde el almacenamiento de la información hasta la presentación de la misma, verificándose que los datos insertados fuesen los mismos que los mostrados en las vistas. Para ejecutar esta prueba, se deben tener datos confiables y sin existencia de duplicidad. La información se midió con los criterios siguientes:

Eficiente: El conjunto de datos de entrada debe ser el mismo que el de salida.

Parcial: El conjunto de datos de salida es el 80 % del conjunto de datos de entrada.

Deficiente: El conjunto de datos de salida es menor del 80 % del conjunto de datos de entrada.

El resultado obtenido fue: Eficiente.

Fue necesario el almacenamiento y comprobación de la información para lograr que los estudiantes trabajaran con datos confiables, y resolvieran las tareas docentes a través de medios correctos y no erróneos. Esto se pudo comprobar una vez realizada la prueba de validez y verificación.

3.8.2 Resultado de la prueba del software.

Con los datos insertados del epígrafe anterior se realizó correctamente la prueba a los estudiantes del grupo 7302 de la Facultad 7, se pudo comprobar en la base de datos los estilos de aprendizaje de cada estudiante registrado y como resolvían las tareas con los medios que le brindaba el sistema, verificando siempre si eran los correctos para su(s) estilo(s) de aprendizaje. En las tablas mostradas a continuación se encuentran los datos almacenados una vez realizada la prueba.

En correspondencias con lo expresado anteriormente en la base de datos se registraron correctamente los estudiantes encuestados, resultado mostrado en la tabla **estudiantes** y la tabla **estudiante_estilo_aprendizaje**.

Representación de la tabla **estudiantes** de la base de datos.

Usuario	Nombre del estudiante
aaarias	Alberto Alejandro Arias Benítez
adumenigo	Alexy Dumenigo Águila
aemena	Adrian Ezequiel Mena Rodríguez
aporras	Adrian Porras Cabrera
caguilera	Claudia Aguilera Pérez

ccvaillant	Cristina De La Caridad Vaillant Valdez
cjheredia	Claudia Jiménez Heredia
egonzalez	Erdin Espinosa Gonzalez
enapoles	Edney Nápoles Mejías
hmvalcarcel	Héctor Manuel Valcarcel Blanco
jjgrass	Jairo José Grass Portelles
jmleal	Juan Miguel Pérez Leal
lcrego	Landy Torres Crego
ljmoleiro	Lázaro Javier Pérez Moleiro
lmcorra	Luis Manuel Cruz Correa
maleyva	Manuel Alejandro López Leyva
mrmoreno	Maidevis Rodríguez Moreno
rochoa	Ramón Ochoa Guerra
yihidalgo	Yelena Isabel Hidalgo Céspedes
ylavalle	Yuliet Fernández Lavalle
ympaz	Yoan Manuel Paz Gómez
ylopezg	Yolexis López Gonzalez
yabalos	Yanet Gutiérrez Avalos
rpeguero	Rosalba Rivera Peguero
nsegura	Naryara Segura Rodríguez
isoria	Liana Soria Berges
dmonte	Dalain Montes Madrigal
lmsoto	Luis Miguel Álvarez Soto
lnavarro	Lisandra Navarro Peguero
lolivarez	Lisandra Olivares Labarcena

Representación de la tabla estudiante_estilo_aprendizaje de la base de datos.

Estilo de aprendizaje:

1. Visual	2. Verbal	3. Global	4. Analítico
5. Planificado	6. Espontáneo	7. Cooperativo	8. Independiente

usuario	Estilo de aprendizaje
aaarias	1 – 2 – 3 – 4 – 6 – 8
adumenigo	1 – 3 – 4 – 5 – 7 – 8
aemena	1 – 3 – 5 – 6
aporras	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8
caguilera	1 – 2 – 3 – 4 – 5 – 8
ccvillant	1 – 3 – 4 – 5 – 7 – 8
cjheredia	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8
egonzalez	1 – 2 – 3 – 5 – 6 – 7 – 8
enapoles	1 – 3 – 4 – 5 – 6
hmvalcarcel	1 – 2 – 3 – 4 – 5 – 6 – 7
jjgrass	1 – 2 – 3 – 4 – 5 – 6 – 7
jmleal	1 – 2 – 3 – 4 – 5 – 8
lcrego	1 – 3 – 5 – 7
ljmoleiro	1 – 3 – 7
lmcorra	1 – 2 – 3 – 4 – 5 – 7 – 8
maleyva	1 – 2 – 3 – 5 – 6 – 7
mrmoreno	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8
rochoa	1 – 2 – 3 – 4 – 5 – 7
yihidalgo	1 – 2 – 3 – 5 – 6 – 8
ylavalle	1 – 2 – 3 – 4 – 5 – 7 – 8
ympaz	2 – 3 – 4 – 5 – 6 – 8
ylopezg	1 – 2 – 3 – 7 – 8
yabalos	1 – 2 – 6 – 7
rpeguero	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8
nsegura	1 – 3 – 4 – 5 – 7 – 8
isoria	1 – 2 – 4 – 5 – 7
dmonte	1 – 2 – 3 – 5 – 7
lmsoto	1 – 2 – 3 – 5 – 6 – 7 – 8
lnavarro	1 – 3 – 5 – 6
lolivarez	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8

En epígrafes anteriores se ha explicado el uso de la técnica de IA razonamiento basado en casos, en la prueba realizada al software se comprobó el uso de la información contenida en la base de datos, el resultado se representa en la tabla siguiente.

Representación de la tabla caso de la base de datos.

id_caso	rasgo predictor	habilidad_especifica_id_habe
1	1	2
2	1	1
3	3	3
4	6	2
5	14	1
6	13	2
7	11	2
8	11	1
9	15	2
10	16	2
11	19	1
12	13	5
13	9	1
14	19	3
15	19	4
16	20	1
17	20	2
18	20	3
19	20	4
20	20	6
21	20	5
22	1	3
23	1	4
24	1	5
25	1	6
26	21	1

Consideraciones parciales.

Con el desarrollo de este capítulo queda explicado el patrón de arquitectura y diseño a utilizar para el desarrollo del sistema. Mediante las tarjetas CRC quedan expuestas las principales clases asociadas en la estructura de la aplicación y se construyó el modelo de datos, el diagrama de despliegue los cuales sirvieron de guía para la implementación del sistema. Con las pruebas realizadas a cada una de las funcionalidades se comprobó el buen funcionamiento de la aplicación y el cumplimiento de las descripciones en las historias de usuario descritas en el capítulo anterior, dando respuesta a las tareas de la investigación.

Conclusiones generales

Luego de finalizada la investigación referida a la implementación del sistema tutorial inteligente para la asignatura de Ingeniería de Software 2 se concluye:

- ✓ El estudio de sistemas tutoriales inteligentes como CircSim, AGT, STIITS y HESEI determinó que estos no ofrecen una solución a lo que se necesita en el tutorial inteligente para la personalización de las tareas docentes teniendo en cuenta las características de los estudiantes, ya que sus funcionalidades no eran adaptables a las deseadas. La investigación sobre estos sistemas aportó un acercamiento a las tendencias mundiales en cuanto al desarrollo de tutoriales inteligentes.
- ✓ El estudio y aplicación de la metodología XP constituyó una práctica eficiente que permitió la obtención exitosa de los artefactos generados, muy pocos pero importantes en cada una de las fases transitadas en el desarrollo del sistema, lo cual fue de gran ayuda en la obtención del producto final.
- ✓ La investigación sobre las ventajas y desventajas de las herramientas propuestas confirmó que eran las indicadas para el desarrollo del trabajo, pues ofrecieron los beneficios necesarios para la implementación de la aplicación.
- ✓ La obtención de los artefactos generados por la metodología XP en el desarrollo de software permitió puntualizar los requisitos funcionales y no funcionales que debía cumplir el sistema tutorial inteligente, para dar respuesta a las necesidades planteadas en la asignatura Ingeniería de Software 2. Se identificaron y diseñaron estas funcionalidades, implementando el sistema con un mínimo margen de error, evitando de esta manera la pérdida de tiempo y recursos.

Recomendaciones

Para futuras versiones del sistema se recomienda:

Incorporar en la base de datos varias características de los estudiantes que hagan el sistema aún más fuerte, confiable en su respuesta y con mejores resultados, la estrategia de aprendizaje del alumno, sus evaluaciones, su estado motivacional, el perfil de inteligencia y el estilo de aprendizaje trabajado en el sistema permitirá tomar decisiones mucho más cercanas al pensamiento humano, objetivo principal de las técnicas de la Inteligencia Artificial.

Agregar diferentes preguntas al cuestionario de satisfacción para enriquecer el resultado obtenido en la respuesta de los estudiantes y propiciar una respuesta por parte del sistema que permita darle seguimiento a los planteamientos tratados.

Extender el uso y desarrollo de la herramienta para los temas definidos en el plan de estudio de la carrera en la Universidad de las Ciencias Informáticas.

Bibliografía y Referencias bibliográficas

1. **González Planas, Ignacio.** *Discurso en la segunda fase de la cumbre Mundial sobre la Sociedad de la Información.* Cuba : s.n., 2005.
2. **A Guevara, David Leonardo.** *Modelo para el desarrollo de Software Educativos.* Cuba : s.n., 2005.
3. **Portal UCI.** Portal de la Universidad de las Ciencias Informáticas. [En línea] [Citado el: 1 de abril de 2013.] <http://www.uci.cu/pregrado>.
4. **Horruitiner Silva, Pedro.** La universidad cubana: el modelo de formación. [En línea] 2006. [Citado el: 2 de abril de 2013.] <http://revistas.mes.edu.cu/greenstone/collect/repo/import/repo/20120706/9789591617989.pdf>.
5. **Ciudad Ricardo, Febe Ángel y Soto López, Nilet.** UCIENCIA. [En línea] 2006. [Citado el: 11 de abril de 2013.] <http://www.monografias.com/trabajos-pdf/ensenanza-ingenieria-software/ensenanza-ingenieria-software.pdf>.
6. Portal de la UCI. [En línea] 2012. <http://www.uci.cu/pregrado#estudiantes>.
7. **Rivera Peguero, Rosalba , Nápoles Mejías, Edney y Tejera Hernández, Dayana Caridad.** *CONSIDERACIONES SOBRE EL DISEÑO DE TUTORIALES INTELIGENTES PARA PERSONALIZAR LAS TAREAS DOCENTES DE IS2.* Ciudad de la Habana : s.n., 2013.
8. **Fernández González, Ana María.** *Estilos de comunicación. La comunicación y su importancia en la educación.* La Habana : Félix Varela, 2006. 165-177.
9. **Cabrera Álvarez, J S.** *Encuentro. Revista de investigación e innovación en la clase de lenguas.* 2005. 14-24.
10. **Meneses Benítez, Gerardo .** El proceso de enseñanza aprendizaje. [En línea] 2007. [Citado el: 1 de abril de 2013.] <http://www.tdx.cat/bitstream/handle/10803/8929/Elprocesodeensenanza.pdf>.
11. El Proceso de Enseñanza Aprendizaje. [En línea] 1999. [Citado el: 11 de marzo de 2013.] <http://www.infor.uva.es/~descuder/docencia/pd/node24.html>.
12. **Parra O, Franklin .** Sistema Tutorial inteligente. [En línea] 2007. [Citado el: 11 de marzo de 2013.] [http://www.dspace.espol.edu.ec/bitstream/123456789/13617/1/Sistema Tutorial Inteligente.pdf](http://www.dspace.espol.edu.ec/bitstream/123456789/13617/1/Sistema%20Tutorial%20Inteligente.pdf).
13. *Preparación pedagógica integral para profesores universitarios.* María Eugenia de la Vega García. Ciudad de la Habana : Félix Varela, 2003. págs. 22-23.
14. El software educativo un medio de enseñanza eficiente. [En línea] julio de 2011. [Citado el: 1 de abril de 2013.] <http://www.eumed.net/rev/ced/29/sml.html>.
15. **Vidal Ledo, María, Gómez Martínez, Freddy y Ruiz Piedra, Alina M.** Software educativos. [En línea] 2010. [Citado el: 1 de abril de 2013.] <http://scielo.sld.cu/pdf/ems/v24n1/ems12110.pdf>.

16. el software educativo. [En línea] 2006. [Citado el: 1 de abril de 2013.]
http://cursa.ihmc.us/rid=1196862742453_516504673_8298/SOFTWARE_EDUCATIVO.pdf.
17. Software Educativo: Software, Definición y Características. [En línea] 2007. [Citado el: 1 de abril de 2013.]
<http://tecno-educativa.blogspot.com/2007/03/software-definicion-y-caractersticas.html>.
18. STI. UN APORTE DE LA INTELIGENCIA ARTIFICIAL PARA LA MEDIACIÓN PEDAGÓGICA. [En línea] 2004. [Citado el: 14 de marzo de 2013.] <http://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/view/279/528>.
19. Inteligencia Artificial Concepto, Características y Metodologías de La Inteligencia Artificial. [En línea] 2008.
<http://inteligenciaartificialudb.blogspot.com/2008/01/concepto-caractersticas-y-metodologas.html>.
20. Inteligencia Artificial (IA). [En línea] 2005.
http://www.profesores.frc.utn.edu.ar/industrial/sistemasinteligentes/FFlexible/Inteligencia_Artificial.pdf.
21. **Lage, Fernando J y Cataldi, Zulma.** Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia. [En línea] 2009. [Citado el: 11 de abril de 2013.]
<http://www.virtualeduca.info/ponencias2009/91/Modelo%20STI%20Internet.doc>.
22. Sistemas Tutoriales Inteligentes (STI). [En línea] junio de 2012. [Citado el: 11 de abril de 2013.]
<http://www.buenastareas.com/ensayos/Sistemas-Tutoriales-Inteligentes/4558869.html>.
23. **Rojas Correa, Yharllan Alex y Muñoz, Teodulo Alfredo.** MENTOR:Sistema Tutorial Inteligente para el desarrollo de habilidades en la solución de problemas matemáticos. [En línea] 2007. [Citado el: 11 de marzo de 2013.] <http://redalyc.uaemex.mx/pdf/952/95270210.pdf>.
24. **Pérez Gama, Alfonso.** Sistema de Tutoriales Inteligentes. Diseño de sistemas expertos en la Educación. [En línea] 2006. [Citado el: 11 de marzo de 2013.]
http://www.dspace.espol.edu.ec/bitstream/123456789/13617/1/Sistema_Tutorial_Inteligente.pdf.
25. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia. Módulo del Estudiante. [En línea] 2009. <http://www.virtualeduca.info/ponencias2009/91/Modelo%20STI%20Internet.doc>.
26. Modelo de Sistemas Tutor Inteligente distribuido para educación a distancia. Módulo de Dominio. [En línea] 2009. <http://www.virtualeduca.info/ponencias2009/91/Modelo%20STI%20Internet.doc>.
27. Sistemas Tutoriales Inteligentes. Un análisis crítico. [En línea] 9 de febrero de 2009.
http://postgrado.info.unlp.edu.ar/Carreras/Especializaciones/Tecnologia_Informatica_Aplicada_en_Educacion/Trabajos_Finales/Huapaya.pdf.
28. [En línea] 9 de febrero de 2009. [Citado el: 11 de marzo de 2013.]
http://postgrado.info.unlp.edu.ar/Carreras/Especializaciones/Tecnologia_Informatica_Aplicada_en_Educacion/Trabajos_Finales/Huapaya.pdf.
29. El Razonamiento basado en casos.RBC. [En línea] enero de 2011. [Citado el: 11 de abril de 2013.]
<http://www.buenastareas.com/ensayos/El-Razonamiento-Basado-En-Casos/1418945.html>.

30. Razonamiento basado en casos. Una visión general. [En línea] 2009. [Citado el: 11 de abril de 2013.] <http://es.scribd.com/doc/100017635/Razonamiento-Basado-en-Casos>.
31. UN SISTEMA BASADO EN CASOS PARA LA TOMA DE DECISIONES EN CONDICIONES DE INCERTIDUMBRE. [En línea] 2002. [Citado el: 11 de abril de 2013.] <http://rev-inv-ope.univ-paris1.fr/files/23202/IO-23202-1.pdf>.
32. *Sistemas Tutores Inteligentes Orientados a la Enseñanza para la Comprensión*. **Cataldi, Zulma**. Facultad Regional de Buenos Aires. Argentina : in EDUTEC Revista Electrónica de Tecnología Educativa, 28 de marzo de 2009.
33. **García García, Maribel, León Vidal, Lisset y Toledo Rivero, Viviana R.** STIITS, Sistema Tutorial Tnteligente para Diagnostico y Tratamiento de Infecciones de Transmisión Sexual. [En línea] 2009. [Citado el: 13 de marzo de 2013.] http://informatica2009.sld.cu/Members/mggarcia/stiits-sistema-tutorial-tnteligente-para-diagnostico-y-tratamiento-de-infecciones-de-transmision-sexual/at_download/trabajo.
34. Revista Electrónica de Tecnología Educativa. [En línea] marzo de 2009. [Citado el: 13 de marzo de 2013.] <http://edutec.rediris.es/Revelec2/revelec28/>.
35. **Martínez Sánchez, Natalia, y otros, y otros.** *Representación y organización del conocimiento en Software educativo*. 2009.
36. **López Barrio, C.** Metodología de Desarrollo: Programación Extrema. [En línea] 2006. [Citado el: 19 de marzo de 2013.] http://www-lsi.die.upm.es/carreras/ISSE/programacion_extrema_1.x2.pdf.
37. **Cleger Despaigne, Eliober y Tornés Montes de Oca, Annarella María.** *Análisis y Diseño de una herramienta interactiva de simulación de procesos: Nodo Virtual de Procesos. Segunda Iteración. Tesis (ingeniero en ciencias informáticas)*. Ciudad de La Habana(UCI) : s.n., 2009.
38. **Letelier, Patricio y Penadés, María Carmen.** Metodologías ágiles para el desarrollo de Software: (XP). [En línea] 2006. [Citado el: 19 de marzo de 2013.] <http://www.willydev.net/descargas/masyxp.pdf>.
39. Metodologías ágiles para el desarrollo de software. XP. [En línea] 2006. <http://www.willydev.net/descargas/masyxp.pdf>.
40. **Letelier, Patricio y Penadés, Carmen.** *Metodologías ágiles para el desarrollo de software: Extreme Programming (XP)*. Valencia : s.n.
41. Extreme Programmig Explained: Notas sobre Metodologías Ágiles. [En línea] 2009. [Citado el: 19 de marzo de 2013.] <http://migueljaque.com/index.php/metodologias/xp/29-xp/63-programacionextrema>.
42. **González Barbone, Víctor A.** XP: Programación Extrema. Instituto de Ingeniería Eléctrica. Facultad de Ingeniería. [En línea] 2008. [Citado el: 19 de marzo de 2013.] <http://iie.fing.edu.uy/~nacho/blandos/seminario/XProg1.html>.
43. **Echeverry Tobón, Luis Miguel y Delgado Carmona, Luz Elena.** *Caso práctico de la metodología ágil XP al desarrollo de software. Planificación*. 2007. págs. 32-33.

44. **Echeverry Tobón, Luis Miguel y Delgado Carmona, Luz Elena.** *Caso práctico de la metodología ágil XP al desarrollo de software. Diseño.* 2007. págs. 36-38.
45. **Echeverry Tobón, Luis Miguel y Delgado Carmona, Luz Elena.** *Caso práctico de la metodología ágil XP para el desarrollo de software. Implementación.* 2007. págs. 38-40.
46. —. *Caso práctico de la metodología ágil XP para el desarrollo de Software. Pruebas.* 2007. págs. 41-42.
47. **Pérez González, Rodrigo y Rodríguez Martín, Aureliano David.** Metodología de desarrollo del Software. [En línea] 9 de octubre de 2008. [Citado el: 1 de abril de 2013.] <http://solusoft-g11.googlecode.com/files/Metodologias%20de%20desarrollo.pdf>.
48. Metodología RUP. [En línea] 2012. [Citado el: 11 de abril de 2013.] <http://creatividadytecnologia.com/web/wp-content/uploads/2012/07/METODOLOGIA-RUP.doc>.
49. **Cabrera González, Lianet y Pompa Torres, Enrique Roberto.** Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información. [En línea] 15 de octubre de 2012. [Citado el: 12 de abril de 2013.] <http://publicaciones.uci.cu/index.php/SC/article/download/1032/581>.
50. **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico.* Quinta. 2010. 8448132149.
51. Introducción a Rational Rose. [En línea] 2006. [Citado el: 1 de abril de 2013.] <http://www.essi.upc.edu/~ese/web/documents/lab/0304Q2/lessons/lese-2/LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt>.
52. Innovar para crecer. [En línea] 2007. [Citado el: 12 de abril de 2013.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
53. **Belmonte Fernández, Oscar.** Introducción al lenguaje de programación Java. [En línea] 2005. [Citado el: 25 de mayo de 2013.] <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>.
54. Características del lenguaje Java. [En línea] 2000. [Citado el: 1 de abril de 2013.] <http://www.iec.csic.es/cryptonicon/java/quesjava.html>.
55. Desventajas de Java. [En línea] 2006. <http://meetingjava.blogspot.com/2006/08/ventajas-y-desventajas-del-java.html>.
56. PHP. Lenguaje de Programación. [En línea] 29 de enero de 2010. [Citado el: 12 de abril de 2013.] <http://programandoideas.com/el-lenguaje-de-programacion-php/>.
57. lenguaje PHP. [En línea] 2008. [Citado el: 12 de abril de 2013.] <http://redgrafica.com/El-lenguaje-de-programacion-PHP>.
58. Principios para el lenguaje PHP. [En línea] 17 de enero de 2013. [Citado el: 12 de abril de 2013.] <http://www.desarrolloweb.com/manuales/12/>.
59. Ventajas de por qué elegir PHP. [En línea] 2008. http://www.programacion.com/articulo/por_que_elegir_php_143.

60. Que es una sistema gestor de base de datos. [En línea] 2007. [Citado el: 12 de abril de 2013.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
61. Las principales características de mysql. [En línea] 2007. [Citado el: 12 de abril de 2013.] <http://dev.mysql.com/doc/refman/5.0/es/features.html>.
62. Las principales características de MySQL. [En línea] 2007. [Citado el: 1 de abril de 2013.] <http://dev.mysql.com/doc/refman/5.0/es/features.html>.
63. Postgresql vs Mysql. [En línea] 2007. [Citado el: 12 de abril de 2013.] http://danielpecos.com/docs/mysql_postgres/x15.html.
64. PostgreSQL. [En línea] 2 de octubre de 2010. [Citado el: 12 de abril de 2013.] http://www.postgresql.org.es/sobre_postgresql.
65. PostgreSQL. Su historia. [En línea] 2006. [Citado el: 1 de abril de 2013.] http://danielpecos.com/docs/mysql_postgres/x15.html.
66. SISTEMA PARA EL CONTROL DEL USO DE LOS SOFTWARES EDUCATIVOS. [En línea] 2009. [Citado el: 2 de abril de 2013.] <http://www.eumed.net/libros-gratis/2009c/585/Descripcion%20del%20modelo%20del%20dominio.htm>.
67. Tecnología y Synergix. [En línea] 2008. [Citado el: 20 de abril de 2013.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
68. **Sommerville, Ian**. Requisitos del Software. [En línea] 2004. [Citado el: 2 de abril de 2013.] <http://lsi.ugr.es/~ig1/docis/requeintro.pdf>.
69. Ciclo de vida de un proyecto XP. [En línea] 2006. [Citado el: 1 de mayo de 2013.] <http://oness.sourceforge.net/proyecto/html/ch05s02.html>.
70. Metodologías SCRUM y XP. [En línea] 2010. http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_SCRUM_y_XP#FASES_DE_LA_METODOLOG.C3.8DA_XP.
71. Fases de la Programación Extrema. [En línea] 2008. [Citado el: 5 de abril de 2013.] <http://programacionextrema.tripod.com/fases.htm>.
72. Actividades de implementación Tarjetas CRC. [En línea] 10 de noviembre de 2004. [Citado el: 5 de abril de 2013.] http://www.inf.utfsm.cl/~visconti/xp/Tarjetas_CRC_2.doc.
73. **Veloso Hernández, Pedro**. Uso de patrones de arquitectura. [En línea] 2009. [Citado el: 5 de abril de 2013.] <http://jjegonzalezf.files.wordpress.com/2009/07/uso-de-patrones-de-arquitectura-capitulo-4.ppt>.
74. Arquitectura de Software. [En línea] 22 de marzo de 2011. [Citado el: 5 de abril de 2013.] <http://es.scribd.com/doc/51339509/Arquitectura-de-software>.

75. SG. *Evaluando la Arquitectura de Software: Parte 1. Panorama General*. [En línea] 2007. [Citado el: 5 de abril de 2013.] <http://sg.com.mx/content/view/235>.
76. El patrón modelo vista controlador. [En línea] 2008. [Citado el: 6 de abril de 2013.] <http://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf>.
77. MVC. Modelo Vista Controlador. [En línea] 2008. [Citado el: 6 de abril de 2013.] <http://www.desarrolloweb.com/wiki/mvc-modelo-vista-controlador.html>.
78. Protocolo ligero de acceso a directorios (LDAP). [En línea] 2009. [Citado el: 6 de abril de 2013.] <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ldap.html>.
79. El Modelo de Datos. [En línea] 28 de septiembre de 2012. [Citado el: 5 de abril de 2013.] <http://www.slideshare.net/csalazarc/modelo-de-datos-14506949>.
80. Análisis y Diseño de Sistemas II. Diagrama de Despliegue. [En línea] 3 de diciembre de 2008. [Citado el: 5 de abril de 2013.] <http://virtual.usalesiana.edu.bo/web/practica/archiv/desplieg1.ppt>.
81. La prueba de aceptación. [En línea] 2010. [Citado el: 21 de abril de 2013.] <http://pruebasdesoftware.com/pruebadeaceptacion.htm>.

Anexos

Anexos 1.

La encuesta para la obtención del estilo de aprendizaje es predefinida por la universidad y aplicada a los estudiantes de primer año, se encuentra en el sistema de encuestas mediante la dirección <http://encuesta.uci.cu>.

(4)_Cuestionario de Auto-reporte de Estilos de Aprendizaje

¿Te has preguntado alguna vez acerca de las formas que tienes para aprender? ¿Has notado que a veces aprendes los contenidos mucho más rápido cuando se presentan de determinada manera? ¿Has notado que en ocasiones prefieres estudiar solo o muchas veces necesitas intercambiar con varias personas mientras estudias? ¿Has pensando alguna vez en optimizar tu aprendizaje? Respondiendo con sinceridad este cuestionario, podrás dar respuesta a varias interrogantes sobre la forma en que aprendes, lo cual te permitirá conocerte mejor y por consiguiente, desarrollar a un nivel mucho más óptimo tu aprendizaje.

Seleccione:

- 1.- Si la situación no se ajusta en ninguna manera a tus formas preferidas de aprender.
- 2.- Si la situación se ajusta solamente en algunos casos a tus formas preferidas de aprender.
- 3.- Si la situación se ajusta completamente a tus formas preferidas de aprender.

*** 1. Puedo recordar mucho mejor los contenidos cuando los veo escritos.**

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

*** 2. Me siento mucho mejor aprendiendo lo que puedo ver, a tener que aprender de lo que solamente escucho.**

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

*** 3. Me gusta que los profesores escriban en la pizarra, hagan esquemas, gráficos, muestren láminas, etc.**

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

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

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

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

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

Sección B

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

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

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

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 8. No puedo estudiar solo(a), me gusta hacerlo al menos con una persona, o si puedo escuchar música mientras estudio, mucho mejor.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 9. Me es fácil escuchar y seguir al profesor cuando explica en clases, lo comprendo todo, aún cuando no se escriba nada en la pizarra.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

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

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

Sección C

- * 11. Cuando estudio, me concentro mucho más en lo concreto, en lo general, que en las descripciones de los detalles.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 12. Cuando leo materiales lo hago a saltos, buscando solo la información más general e importante, trato de obviar los detalles o la información insignificante.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 13. Cuando estoy en clases me siento mucho mejor cuando tomo notas breves sintetizando todo lo que dice el profesor, que cuando tengo que escribir todo, hasta el más mínimo detalle, de lo que el profesor explica.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 14. Prefiero ser realista y concreto(a) cuando soluciono alguna tarea.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 15. Cuando soluciono problemas, me gusta mucho aplicar la teoría, ver el problema como un todo.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

Sección D

* 16. Sigo en clases hasta el más mínimo detalle de lo que el profesor explica, y así en detalles lo plasmo en mi libreta.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

* 17. Me gustan mucho las Matemáticas, la Estadística, la Lógica, el procesamiento de datos, etc.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

* 18. La mayoría de las veces soy lento(a) al leer y estudiar, pues me gusta captar con profundidad cada idea y detalle.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

* 19. Me gusta ser meticuroso(a) con mis cosas, no me gusta obviar ni el más mínimo detalle.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

* 20. Al procesar la información que recibo cuando escucho a mi profesor, o cuando leo un texto, me gusta aplicar mi experiencia previa, mi intuición, razonar sobre la misma, etc.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

Sección E

- * 21. No me gusta llegar tarde a clases o tardarme en la entrega de algún trabajo orientado por el profesor.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 22. Me gusta ser muy organizado(a) con mis cosas, libretas, libros, etc.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 23. Soy lento(a) para tomar alguna decisión, me gusta primero analizar cada aspecto antes de lanzarme y hacer lo primero que me venga en mente.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 24. Cuando resuelvo alguna tarea, me gusta concentrarme, ver cada detalle o variante y reflexionar sobre cada paso a seguir.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 25. Antes de resolver alguna tarea, me gusta saber el cómo hacerla y tener los pasos para resolver la misma.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

Sección F

- * 26. Me gusta ser rápido(a) cuando tomo una decisión, no me gusta pensarlo mucho.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 27. Me gusta ser espontáneo(a) al hacer las cosas, no me gusta mucho planificar mis actividades.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 28. Me gusta ser activo(a) en clases, rápidamente me pongo a solucionar las tareas asignadas por el profesor.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
 Se ajusta solamente en algunos casos
 Se ajusta completamente

- * 29. No es habitual en mí pensar mucho lo que voy a responder.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

- * 30. Generalmente muestro gran disposición a la hora de resolver cualquier tarea que se me plantea.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

Sección G

- * 31. Me gusta estudiar en equipo, y para presentarme a exámenes, mucho más.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

- * 32. Me gusta que me asignen responsabilidades o tareas delante de mis otros compañeros de aula.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

- * 33. Me gustan los lugares abiertos, los debates, intercambios, discusiones, etc., donde pueda expresar lo que pienso delante de los demás.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

- * 34. Soy una persona abierta y sociable, me relaciono con los demás con facilidad.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

- * 35. Me aburro mucho cuando estudio solo(a), necesito al menos una persona para intercambiar sobre lo que he aprendido.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

Sección H

* 36. Prefiero las actividades de lectura o de análisis que demanden alta concentración mental, donde me encuentre solo(a), así puedo concentrarme mucho mejor.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

* 37. Para las clases prácticas, talleres, seminarios, exámenes, etc., me gusta prepararme y estudiar solo(a).

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

* 38. Me gustan mucho más las evaluaciones escritas que las orales, donde tengo que expresar mis conocimientos delante de los demás.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

* 39. En las actividades docentes, clases, conferencias, etc., soy generalmente callado(a).

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

* 40. No me concentro cuando tengo que estudiar o trabajar en equipo, me molesta alguien más a mi lado cuando tengo que estudiar.

Seleccione una de las siguientes opciones

- No se ajusta en ninguna manera
- Se ajusta solamente en algunos casos
- Se ajusta completamente

Anexos 2.

Las historias de usuarios dieron paso el desarrollo e implementación de las funcionalidades identificadas en la planificación del trabajo.

Tabla 11. HU_1 Autenticar usuario.

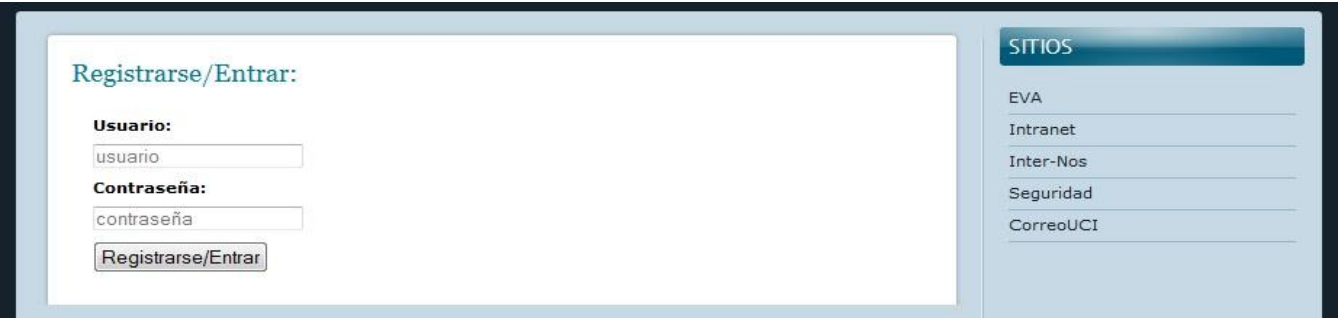
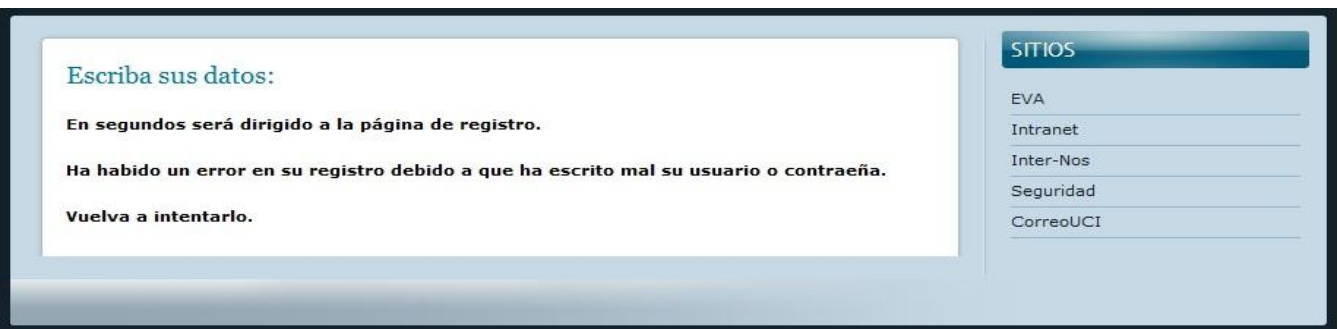
Historia de Usuario	
Número: 1	Nombre: Autenticar usuario.
Usuario: Estudiante, profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 1.	
<p>Descripción: Los estudiantes y profesores deben autenticarse en el sistema para verificar si pertenecen al dominio UCI. Si es estudiante una vez validada la entrada debe realizar la encuesta, si es profesor puede realizar cualquier funcionalidad mostrada en la parte derecha de la página. En caso de incorporar datos erróneos debe mostrarse un mensaje de error.</p>	
<p>Observaciones: Se hace referencia al requisito funcional Autenticar Usuario.</p>	
<p>Autenticar usuario</p> 	
<p>Validar datos del usuario</p> 	

Tabla 12. HU_2 Realizar encuesta de estilo aprendizaje.

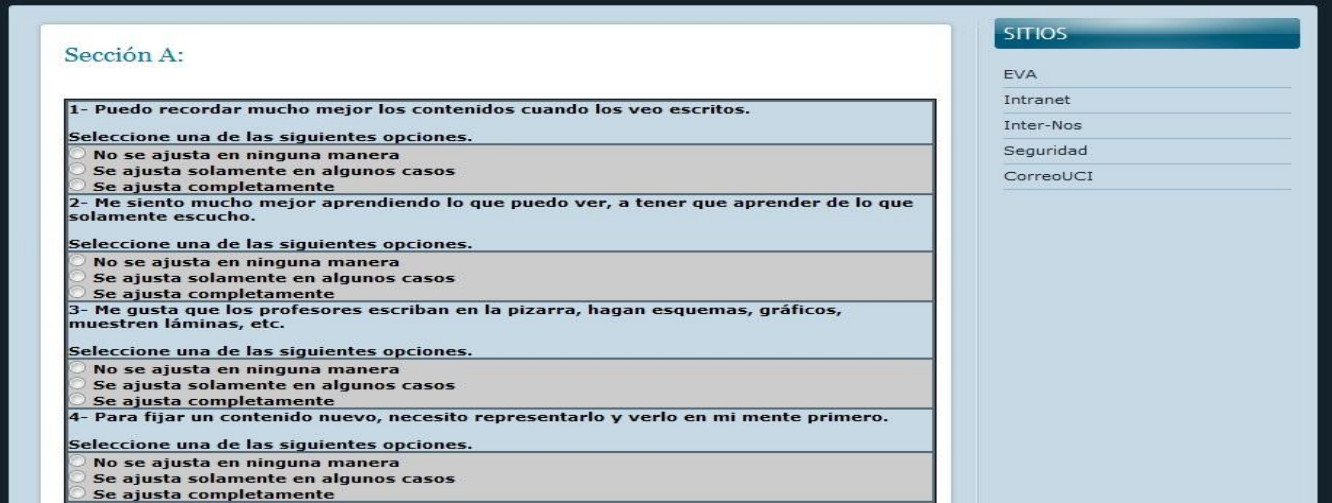
Historia de Usuario	
Número: 2	Nombre: Realizar encuesta de estilo de aprendizaje.
Usuario: Estudiante	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 1.	
<p>Descripción: Los estudiantes una vez autenticados en el sistema podrán realizar la encuesta de estilo de aprendizaje seleccionando las preguntas que lo identifiquen según la forma de percibir la información y procesar los conocimientos. En caso de dejar alguna pregunta sin responder en alguna sección debe mostrarse un mensaje de error y dirigirlo para la página donde dejó de responder la pregunta.</p>	
<p>Observaciones: Se hace referencia al requisito funcional Realizar encuesta de estilo de aprendizaje.</p>	
<p>Sección</p>  <p>The screenshot shows a web interface for a survey. On the left, under 'Sección A:', there are four numbered questions. Each question has three radio button options: 'No se ajusta en ninguna manera', 'Se ajusta solamente en algunos casos', and 'Se ajusta completamente'. The questions are: 1- 'Puedo recordar mucho mejor los contenidos cuando los veo escritos.', 2- 'Me siento mucho mejor aprendiendo lo que puedo ver, a tener que aprender de lo que solamente escucho.', 3- 'Me gusta que los profesores escriban en la pizarra, hagan esquemas, gráficos, muestren láminas, etc.', and 4- 'Para fijar un contenido nuevo, necesito representarlo y verlo en mi mente primero.' On the right side of the screenshot, there is a sidebar titled 'SITIOS' with a list of links: EVA, Intranet, Inter-Nos, Seguridad, and CorreoUCI.</p>	

Tabla 13. HU_3 Registrar estudiante.

Historia de Usuario	
Número: 3	Nombre: Registrar estudiante.
Usuario: Estudiante	


Prioridad en Negocio: alta	Nivel de Complejidad: alta
Iteración Asignada: 1	
Descripción: El estudiante una vez autenticado en el sistema, realizado la encuesta de estilo de aprendizaje podrá realizar sus operaciones en la aplicación culminando así su registro en el sistema.	
Observaciones: Se hace referencia al requisito funcional Registrar estudiante.	
<p>Registro de estudiantes</p> <div style="border: 1px solid black; padding: 5px;"> <p>39- En las actividades docentes, clases, conferencias, etc, soy generalmente llamado(a).</p> <p>Seleccione una de las siguientes opciones.</p> <p><input type="radio"/> No se ajusta en ninguna manera</p> <p><input type="radio"/> Se ajusta solamente en algunos casos</p> <p><input type="radio"/> Se ajusta completamente</p> <p>40- No me encuentro cuando tengo que estudiar o trabajar en equipo, me molesta alguien más a mi lado cuando tengo que estudiar.</p> <p>Seleccione una de las siguientes opciones.</p> <p><input type="radio"/> No se ajusta en ninguna manera</p> <p><input type="radio"/> Se ajusta solamente en algunos casos</p> <p><input type="radio"/> Se ajusta completamente</p> </div> <div style="text-align: right; margin-top: 10px;">  </div>	

Tabla 14. HU_4 Ver listado estudiantes.

Historia de Usuario	
Número: 4	Nombre: Ver listado estudiantes.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 1	
Descripción: Los profesores autenticados en la aplicación podrán ver el listado de estudiantes, en una tabla se mostrará el usuario y el nombre de los estudiantes registrados en el sistema,	
Observaciones: Se hace referencia al requisito funcional Ver listado estudiantes.	

Listado de estudiantes

Estudiantes:

Nombre	Usuario	Eliminar
Adrian Ezequiel Mena Rodriguez	aemena	<input type="checkbox"/>
Edney Napoles Mejias	enapoles	<input type="checkbox"/>

CATEGORÍAS

- Gestión de Material
- Gestión de Asignatura
- Gestión de Tema
- Gestión de Tarea
- Gestión de Habilidad
- Gestión de Habilidad Especifica

Tabla 15. HU_5 Gestionar tarea de estudiantes.

Historia de Usuario	
Número: 5	Nombre: Gestionar tarea de estudiantes.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: alta
Iteración Asignada: 2	
<p>Descripción: La presente historia de usuario tiene como objetivo la gestión de las tareas, debe permitir adicionar tareas y eliminar las tareas que el profesor elabore e inserte en el sistema. Las tareas tienen resultado (nombre de la tarea). El nombre de la tarea será insertado en un campo de texto donde solo podrá escribir letras, en caso contrario mostrar mensaje de error.</p>	
<p>Observaciones: Se hace referencia al requisito funcional gestionar tarea.</p>	

Agregar tareas:

Nombre de la tarea:*

Tareas:

Nombre	Eliminar
tarea número dos	<input type="checkbox"/>
tarea número uno	<input type="checkbox"/>

CATEGORÍAS

- Gestión de Material
- Gestión de Asignatura
- Gestión de Tema
- Gestión de Tarea
- Gestión de Habilidad
- Gestión de Habilidad Especifica
- Gestión de Material Especifico
- Relacionar Habilidad-Tarea
- Evaluar Tarea
- Evaluación Estudiantes
- Encuesta de Satisfacción
- Estudiantes registrados

Tabla 16. HU_6 Asignar materiales para resolver tareas.

Historia de Usuario	
Número: 6	Nombre: Asignar materiales para resolver tareas.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: alta
Iteración Asignada: 3	
Descripción: El estudiante una vez autenticado en el sistema podrá ver las tareas almacenadas en la aplicación, selecciona la tarea que desee realizar y podrá tener la opción de adjuntar la tarea una vez resuelta, en caso de no llegar a su resultado podrá revisar materiales según la habilidad específica que servirán de apoyo y le permitirá llegar a su resultado, en caso de no haber tareas en el sistema mostrar mensaje.	
Observaciones: se hace referencia al requisito funcional Asignar tarea.	

1


En esta tarea usted debe:

Describir y especificar la estructura del sistema a construir


El adjunto de la tarea debe ser nombrado de la siguiente forma usuario_nombre, ejemplo: enapoles_Edney Napoles y debe estar en un archivo compactado.

Adjunto:*


En caso de no poder resolver esta tarea con sus conocimientos actuales, deje que el sistema le brinde algunos materiales de ayuda haciendo click en este botón:



TAREAS



SATISFACCIÓN



SITIOS

EVA
Intranet
Inter-Nos
Seguridad

2

Habilidades específicas a cumplir en la tarea: Realizar diagrama de clases con estereotipos web.

Seleccione una de las habilidades listadas abajo para visualizar los materiales que le ayudarán a resolver la tarea, o entender la habilidad específica seleccionada.

Habilidad específica:*

--Seleccione--


--Seleccione--

Identificar las paginas clientes

Identificar la o las paginas servidoras

Identificar la clase controladora

TAREAS



SATISFACCIÓN

3

Materiales específicos para la habilidad: Identificar las páginas clientes

[archivos/Materiales_Especificos/Presentacion diapositivas 2.pptx en ../archivos/Materiales_Especificos/Presentacion diapositivas 2.pptx](#)


[archivos/Materiales_Especificos/Archivo comprimido 2.rar en ../archivos/Materiales_Especificos/Archivo comprimido 2.rar](#)

[archivos/Materiales_Especificos/documento 2.txt en ../archivos/Materiales_Especificos/documento 2.txt](#)


[archivos/Materiales_Especificos/Imagen 2.jpg en ../archivos/Materiales_Especificos/Imagen 2.jpg](#)

[archivos/Materiales_Especificos/Video 2.wmv en ../archivos/Materiales_Especificos/Video 2.wmv](#)

[archivos/Materiales_Especificos/Audio 2.mp3 en ../archivos/Materiales_Especificos/Audio 2.mp3](#)



SATISFACCIÓN



SITIOS

EVA
Intranet
Inter-Nos
Seguridad
CorreoUCI

Tabla 17. HU_7 Ver tareas.

Historia de Usuario	
Número: 7	Nombre: Ver tareas.
Usuario: Profesor, Estudiante	

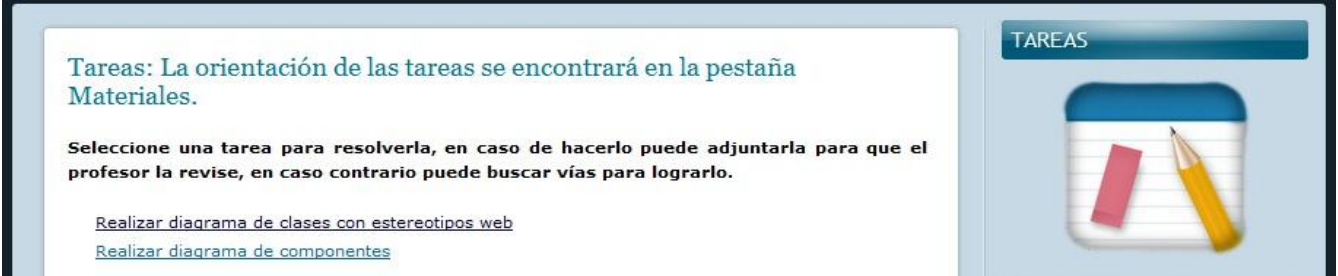
Prioridad en Negocio: media	Nivel de Complejidad: baja
Iteración Asignada: 2	
Descripción: La presente historia de usuario tiene como objetivo que el profesor y el estudiante puedan revisar y observar todas las tareas que hay involucradas en el sistema, en caso de no haber tareas insertadas en la base de datos debe mostrarse un mensaje.	
Observaciones: se hace referencia al requisito funcional Ver tarea.	
Ver tareas estudiante	
	

Tabla 18. HU_8 Subir tareas resueltas.

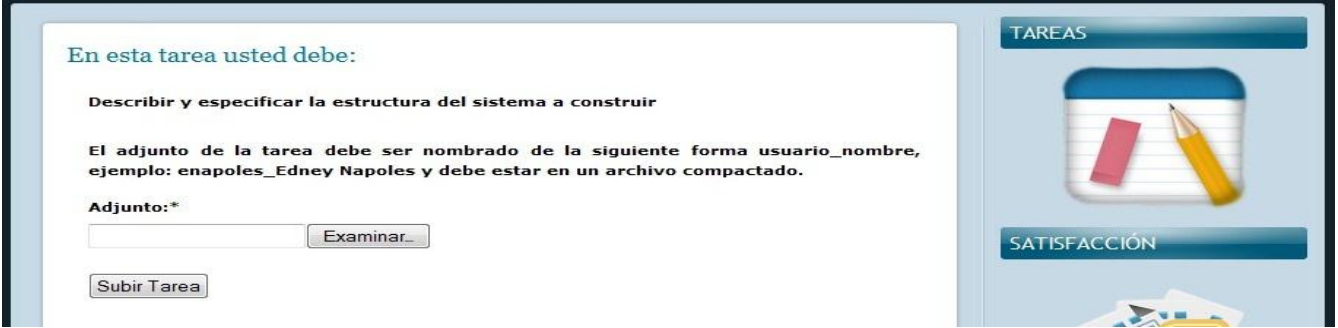
Historia de Usuario	
Número: 8	Nombre: Subir tarea resuelta.
Usuario: Estudiante	
Prioridad en Negocio: alta	Nivel de Complejidad: baja
Iteración Asignada: 3	
Descripción: La presente historia de usuario tiene como objetivo adquirir las tareas resueltas por los estudiantes adjuntándolas en el sistema.	
Observaciones: se hace referencia al requisito funcional subir tarea resuelta.	
	

Tabla 19. HU_9 Evaluar tareas.

Historia de Usuario	
Número: 9	Nombre: Evaluar tareas.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: alta
Iteración Asignada: 3	
Descripción: El profesor es el encargado de revisar las tareas realizadas por el estudiante y darle una evaluación de acuerdo con su desempeño en la misma ya sea de excelente, bien, mal o regular. Las tareas podrán ser evaluadas por las habilidades específicas.	
Observaciones: se hace referencia al requisito funcional Evaluar tareas.	



2

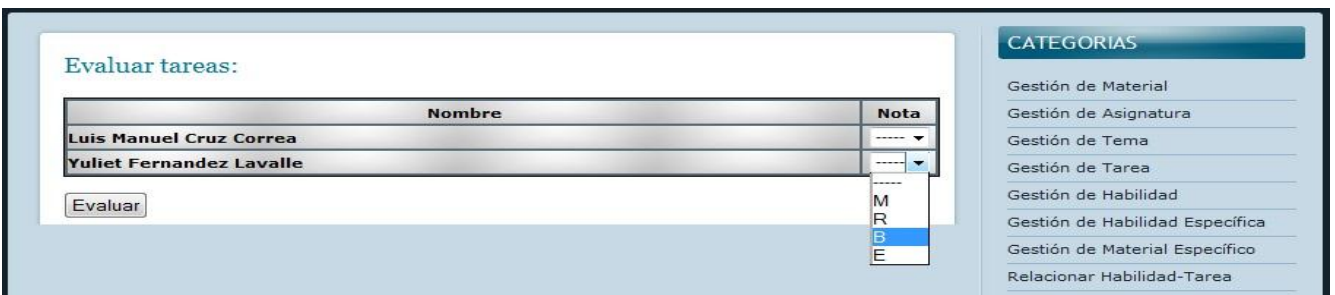


Tabla 20. HU_10 Mostrar evaluación de tareas.

Historia de Usuario	
Número: 10	Nombre: Mostrar evaluación de tareas.
Usuario: Profesor	
Prioridad en Negocio: media	Nivel de Complejidad: media

Iteración Asignada: 3

Descripción: El estudiante debe tener conocimiento de los resultados obtenidos en las tareas a las que le ha dado solución, es por ello que el profesor se encarga de darle a conocer su avance y cuánto ha mejorado en el transcurso de la asignatura, en caso de no haber estudiantes evaluados mostrar mensaje.

Observaciones: se hace referencia al requisito funcional Mostrar evaluación de tareas.

Mostrar evaluación

Ver evaluación de los estudiantes:

Tarea:

- Seleccione--
- Seleccione--
- Realizar diagrama de clases con estereotipos web
- Realizar diagrama de componentes

CATEGORÍAS

- Gestión de Material
- Gestión de Asignatura
- Gestión de Tema
- Gestión de Tarea
- Gestión de Habilidad
- Gestión de Habilidad Específica
- Gestión de Material Específico

2

Evaluaciones para la tarea: Realizar diagrama de clases con estereotipos web

Tarea	Evaluación
Luis Manuel Cruz Correa	R
Yuliet Fernandez Lavalle	B

- Gestión de Habilidad Específica
- Gestión de Material Específico
- Relacionar Habilidad-Tarea
- Evaluar Tarea
- Evaluación Estudiantes
- Encuesta de Satisfacción
- Estudiantes registrados

Tabla 21. HU_11 Gestionar material educativo.

Historia de Usuario	
Número: 11	Nombre: Gestionar material educativo.
Usuario: Profesor	
Prioridad en Negocio: baja	Nivel de Complejidad: media
Iteración Asignada: 2	

Descripción: La presente historia de usuario tiene como objetivo adicionar materiales educativos y eliminar materiales educativos del sistema, servirán de apoyo al estudiante como una vía extra para resolver las tareas, además contará con la orientación de los ejercicios. Tendrá nombre y una dirección de donde será adjuntado. En caso de existir un material con el mismo nombre debe mostrar mensaje de error, el nombre no puede contener caracteres extraños.

Observaciones: se hace referencia al requisito funcional Gestionar material educativo.

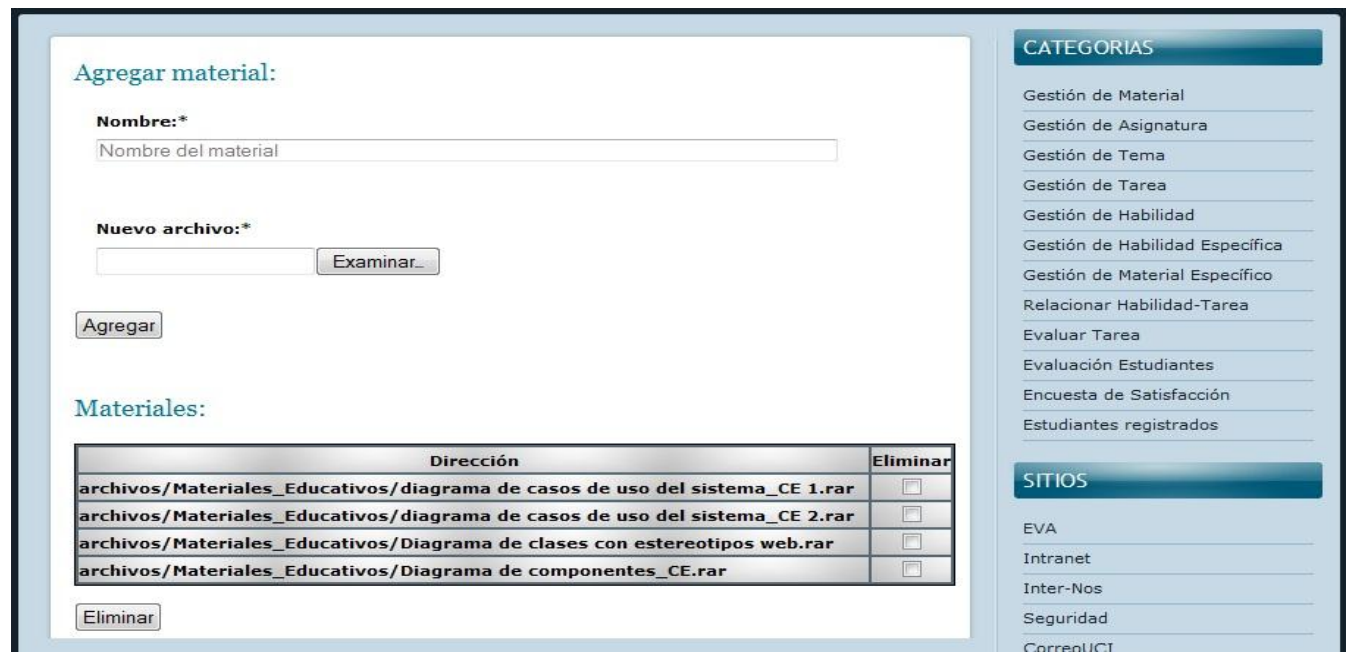


Tabla 22. HU_12 Ver material educativo.

Historia de Usuario	
Número: 12	Nombre: Ver material educativo.
Usuario: Profesor, estudiante	
Prioridad en Negocio: baja	Nivel de Complejidad: media
Iteración Asignada: 2	
Descripción: La presente historia de usuario permitirá mostrar los materiales educativos almacenados en el sistema, podrá abrirlos y descargarlos en caso de ser necesario.	
Observaciones: se hace referencia al requisito funcional Ver material educativo.	

Materiales:

Material 1 en [../archivos/Materiales_Educativos/diagrama de casos de uso del sistema_CE 1.rar](#)

Material 2 en [../archivos/Materiales_Educativos/diagrama de casos de uso del sistema_CE 2.rar](#)

Material 3 en [../archivos/Materiales_Educativos/Diagrama de clases con estereotipos web.rar](#)

Material 4 en [../archivos/Materiales_Educativos/Diagrama de componentes_CE.rar](#)

Gestión de habilidad específica

Gestión de Material Específico

Relacionar Habilidad-Tarea

Evaluar Tarea

Evaluación Estudiantes

Encuesta de Satisfacción

Estudiantes registrados

SITIOS

EVA

Intranet

Inter-Nos

Tabla 23. HU_13 Realizar encuesta de satisfacción.



Historia de Usuario	
Número: 13	Nombre: Realizar encuesta de satisfacción.
Usuario: Estudiante	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 3	
Descripción: La presente historia de usuario permitirá almacenar cómo se siente el estudiante al trabajar con el sistema, guardando los datos recogidos y el estudiante que la realizó.	
Observaciones: se hace referencia al requisito funcional Realizar encuesta de satisfacción.	
<div style="border: 1px solid black; padding: 10px; display: flex; justify-content: space-between;"> <div style="width: 70%;"> <p style="color: #0070c0;">Recomendamos que realice esta encuesta al finalizar la resolución de las tareas:</p> <p>En la encuesta de satisfacción usted podrá opinar como se siente al resolver las tareas que el sistema le orienta y si se siente cómodo con los medios que le brinda para entender los ejercicios. Será un placer registrar y atender su opinión. Gracias.</p> <p>1 - ¿Te sientes cómodo resolviendo las tareas de esta manera?</p> <p><input type="radio"/> Si</p> <p><input type="radio"/> No</p> <p>2 - ¿Te sientes a gusto con los medios que te proporciona el sistema para resolver las tareas?</p> <p><input type="radio"/> Si</p> <p><input type="radio"/> No</p> <p>3 - ¿Qué tipos de medios te gustaría adicionar en el sistema para resolver los ejercicios?</p> <div style="border: 1px solid #ccc; height: 40px; width: 100%;"></div> </div> <div style="width: 25%; border-left: 1px solid black; padding-left: 5px;"> <p style="background-color: #0070c0; color: white; padding: 5px; text-align: center;">TAREAS</p> <div style="text-align: center; margin: 5px 0;">  </div> <p style="background-color: #0070c0; color: white; padding: 5px; text-align: center;">SATISFACCIÓN</p> <div style="text-align: center; margin: 5px 0;">  </div> <p style="background-color: #0070c0; color: white; padding: 5px; text-align: center;">SITIOS</p> </div> </div>	

Tabla 24. HU_14 Mostrar resultado encuesta de satisfacción.



Historia de Usuario	
Número: 14	Nombre: Mostrar resultado encuesta de satisfacción.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 3	
<p>Descripción: La presente historia de usuario tiene como objetivo mostrarles a los profesores el resultado de la encuesta de satisfacción para conocer la opinión del estudiante que realizó la encuesta. En caso de no haber realizado la encuesta mostrar mensaje.</p>	
<p>Observaciones: se hace referencia al requisito funcional Gestionar asignatura.</p>	
<p>Resultado encuesta satisfacción</p>	
	
2	
	

Tabla 25. HU_15 Gestionar asignatura.

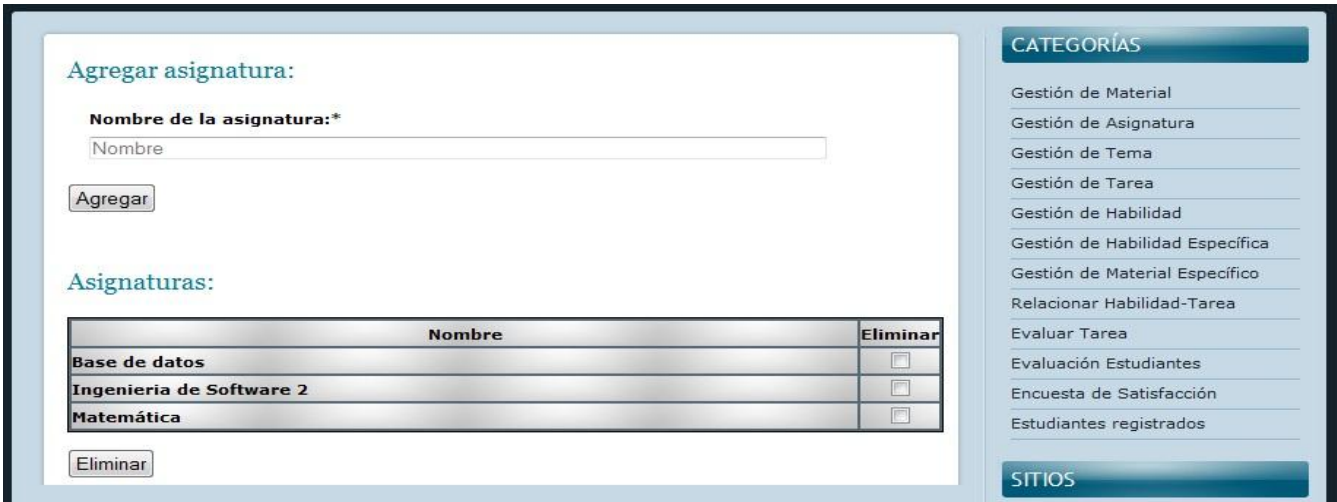
Historia de Usuario	
Número: 15	Nombre: Gestionar asignatura.
Usuario: Profesor	
Prioridad en Negocio: baja	Nivel de Complejidad: media
Iteración Asignada: 1	
Descripción: La presente historia de usuario tiene como objetivo adicionar asignatura y eliminar asignatura del sistema, la asignatura tiene nombre y será insertado mediante un campo de texto donde solo podrán insertarse letras en caso contrario mostrar mensaje.	
Observaciones: se hace referencia al requisito funcional Gestionar asignatura.	
	

Tabla 26. HU_16 Gestionar tema.

Historia de Usuario	
Número: 16	Nombre: Gestionar tema.
Usuario: Profesor	
Prioridad en Negocio: baja	Nivel de Complejidad: media
Iteración Asignada: 1	

Descripción: La presente historia de usuario tiene como objetivo adicionar tema y eliminar tema, se selecciona la asignatura a la que pertenece el tema para adicionar, se escribe el número y el nombre del tema, en caso de existir un tema con el mismo nombre mostrar mensaje.

Observaciones: se hace referencia al requisito funcional Gestionar tema.

Tabla 27. HU_17 Gestionar habilidades.

Historia de Usuario	
Número: 17	Nombre: Gestionar habilidades.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 1	
<p>Descripción: La presente historia de usuario tiene como objetivo adicionar y eliminar habilidad, se escribe el nombre de la habilidad y se selecciona el tema al cual pertenece. Si existe alguna habilidad para el tema seleccionado mostrar mensaje, la habilidad se insertará mediante un campo de texto donde no debe insertarse caracteres diferentes de letras.</p>	
<p>Observaciones: se hace referencia al requisito funcional Gestionar habilidades.</p>	

Tabla 28. HU_18 Gestionar habilidades específicas.

Historia de Usuario	
Número: 18	Nombre: Gestionar habilidades específicas.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 1	
Descripción: La presente historia de usuario tiene como objetivo adicional y eliminar habilidad específica del sistema, se escribe el nombre de la habilidad específica, se selecciona la habilidad y la tarea a la cual pertenece, no se debe insertar un nombre con caracteres extraños, en caso de existir mostrar mensaje.	
Observaciones: se hace referencia al requisito funcional Gestionar habilidades específicas.	

Agregar habilidades específicas:

Nombre de la habilidad específica:*

Nombre de la habilidad general a la que pertenece:*

Nombre de la tarea a la que pertenece:*

Habilidades específicas:

Nombre	Eliminar
Identificar la clase controladora	<input type="checkbox"/>
Identificar las páginas clientes	<input type="checkbox"/>
Identificar la o las páginas servidoras	<input type="checkbox"/>
Identificar correctamente los estereotipos de los componentes	<input type="checkbox"/>
Identificar tipo de relación	<input type="checkbox"/>
Relacionar los componentes lógicamente	<input type="checkbox"/>

CATEGORIAS

- Gestión de Material
- Gestión de Asignatura
- Gestión de Tema
- Gestión de Tarea
- Gestión de Habilidad
- Gestión de Habilidad Especifica
- Gestión de Material Especifico
- Relacionar Habilidad-Tarea
- Evaluar Tarea
- Evaluación Estudiantes
- Encuesta de Satisfacción
- Estudiantes registrados

SITIOS

- EVA
- Intranet
- Inter-Nos
- Seguridad
- CorreoUCI

Tabla 29. HU_19 Gestionar material para las habilidades específicas.

Historia de Usuario	
Número: 19	Nombre: Gestionar material para las habilidades específicas.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 2	
Descripción: La presente historia de usuario tiene como objetivo adicionar y eliminar materiales para las habilidades específicas, tienen un tipo de archivo, una descripción, una dirección donde será adjuntado y se selecciona la habilidad específica a la cual pertenece, en caso de existir el material mostrar un mensaje.	
Observaciones: se hace referencia al requisito funcional Gestionar material para las habilidades específicas.	

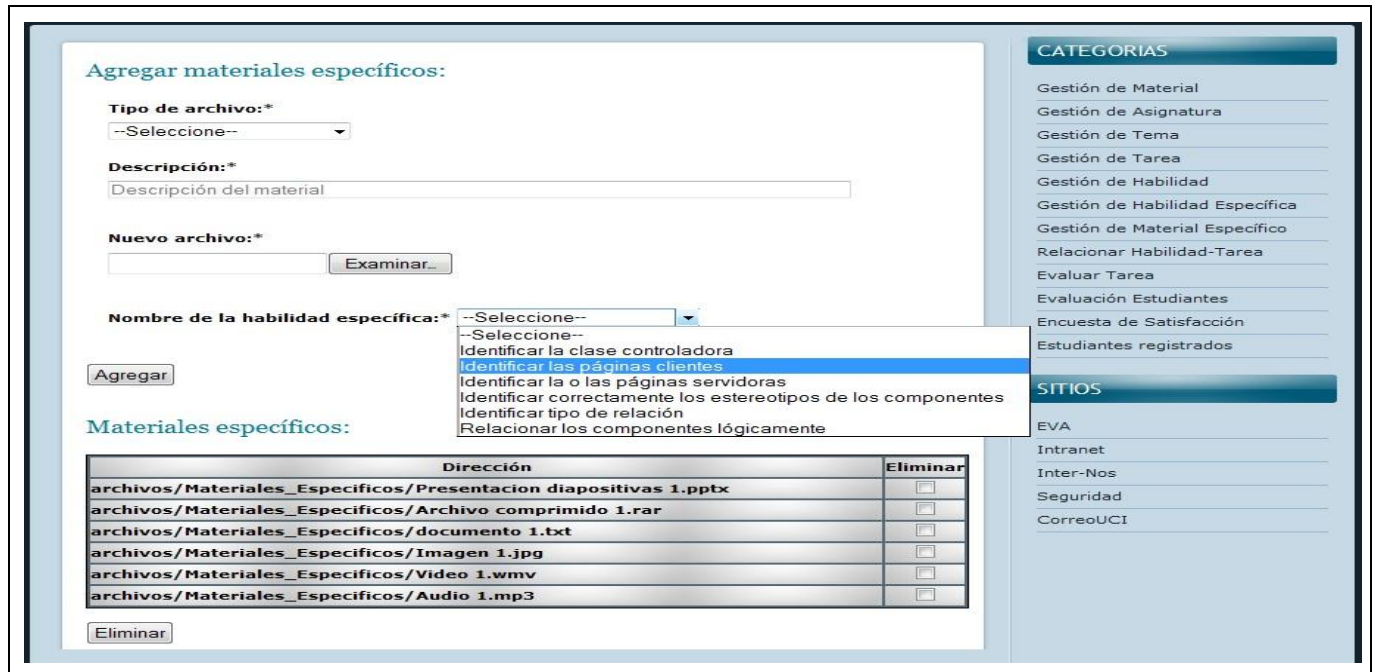


Tabla 30. HU_20 Relacionar habilidad tarea.

Historia de Usuario	
Número: 20	Nombre: Relacionar habilidad tarea.
Usuario: Profesor	
Prioridad en Negocio: alta	Nivel de Complejidad: media
Iteración Asignada: 2	
<p>Descripción: La presente historia de usuario tiene como objetivo relacionar una habilidad a una tarea, para lograrlo debe seleccionar el nombre de la tarea y la habilidad, en caso de no seleccionar algún dato mostrar un mensaje.</p>	
<p>Observaciones: se hace referencia al requisito funcional Relacionar habilidad tarea.</p>	



Tabla 31. HU_21 Mostrar tareas resueltas adjuntadas.

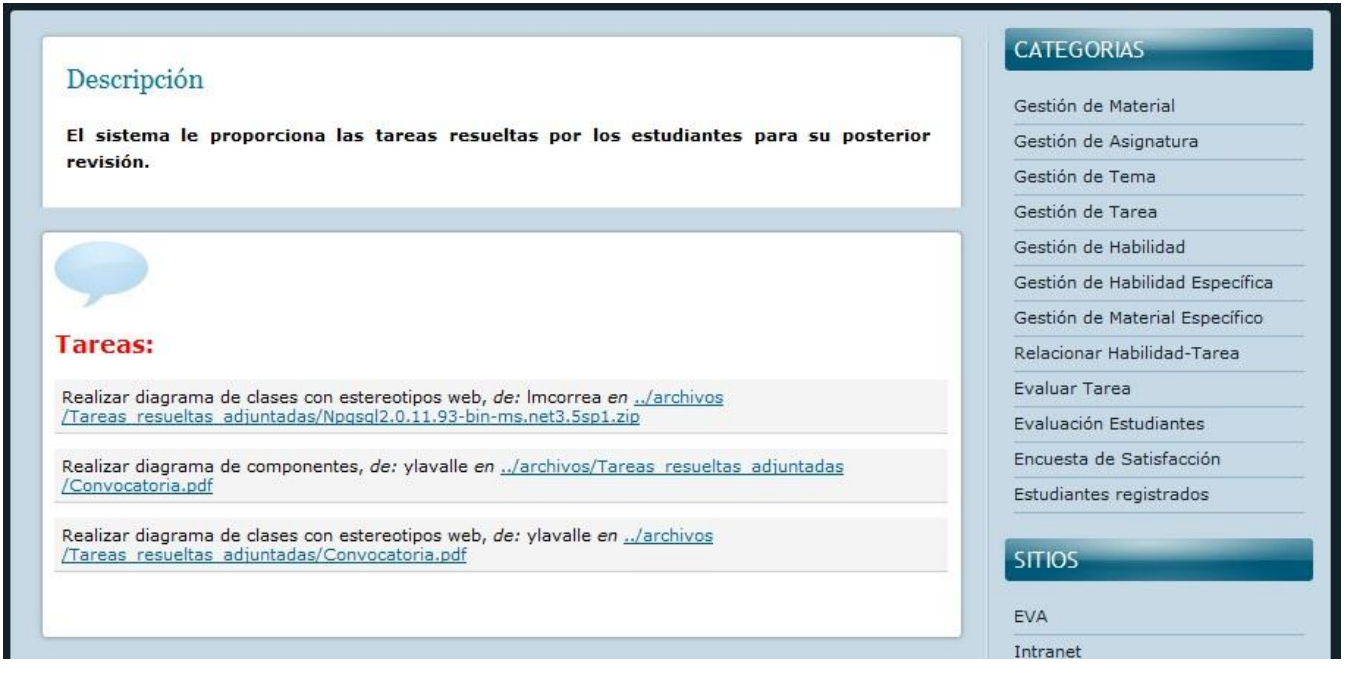
Historia de Usuario	
Número: 21	Nombre: Mostrar tareas resueltas adjuntadas.
Usuario: Profesor	
Prioridad en Negocio: baja	Nivel de Complejidad: media
Iteración Asignada: 3	
Descripción: La presente historia de usuario tiene como objetivo mostrar las tareas adjuntadas en el sistema por los estudiantes para su posterior revisión, debe permitir descargarla y abrirla.	
Observaciones: se hace referencia al requisito funcional Mostrar resultados de encuesta de satisfacción.	
	

Tabla 32. HU_22 Eliminar estudiante.

Historia de Usuario	
Número: 22	Nombre: Eliminar estudiantes.
Usuario: Profesor	
Prioridad en Negocio: media	Nivel de Complejidad: media
Iteración Asignada: 1	

Descripción: La presente historia de usuario tiene como objetivo eliminar estudiantes del sistema, se selecciona el estudiante mediante un checkbox (campo para seleccionar varios atributos) y se elimina de la aplicación, en caso de no existir ningún estudiante mostrar mensaje.

Observaciones: se hace referencia al requisito funcional Mostrar resultados de encuesta de satisfacción.

Estudiantes:

Nombre	Usuario	Eliminar
Alberto Alejandro Arias Benitez	aaarias	<input type="checkbox"/>
Alexy Dumenigo Aguila	adumenigo	<input type="checkbox"/>
Adrian Ezequiel Mena Rodriguez	aemena	<input type="checkbox"/>
Adrian Porras Cabrera	aporras	<input type="checkbox"/>
Claudia Aguilera Perez	caguilera	<input type="checkbox"/>
Cristina De La Caridad Vaillant Valdez	ccvaillant	<input type="checkbox"/>
Claudia Jimenez Heredia	cjheredia	<input type="checkbox"/>
Erdin Espinosa Gonzalez	egonzalez	<input type="checkbox"/>
Edney Napoles Mejias	enapoles	<input type="checkbox"/>
Hector Manuel Valcarcel Blanco	hmvalcarcel	<input type="checkbox"/>
Jairo Jose Grass Portelles	jjgrass	<input type="checkbox"/>
Juan Miguel Perez Leal	jmleal	<input type="checkbox"/>
Landy Torres Crego	lcrego	<input type="checkbox"/>
Lazaro Javier Perez Moleiro	ljmoleiro	<input type="checkbox"/>
Manuel Alejandro Lopez Leyva	maleyva	<input type="checkbox"/>
Maidevis Rodriguez Moreno	mrmoreno	<input type="checkbox"/>
Ramon Ochoa Guerra	rochoa	<input type="checkbox"/>
Yelena Isabel Hidalgo Cespedes	yihidalgo	<input type="checkbox"/>
Yoan Manuel Paz Gomez	ympaz	<input type="checkbox"/>

Eliminar

CATEGORÍAS

- Gestión de Material
- Gestión de Asignatura
- Gestión de Tema
- Gestión de Tarea
- Gestión de Habilidad
- Gestión de Habilidad Específica
- Gestión de Material Especifico
- Relacionar Habilidad-Tarea
- Evaluar Tarea
- Evaluación Estudiantes
- Encuesta de Satisfacción
- Estudiantes registrados

SITIOS

- EVA
- Intranet
- Inter-Nos
- Seguridad
- CorreoUCI

Anexos 3.

Las tareas de la implementación o de ingeniería, se realizaron para resolver las tareas orientadas y darle respuesta al objetivo general.

Tabla 33. TI_1 Describir el sistema propuesto y creación del modelo de dominio.

Tarea de Ingeniería	
Número de la tarea: 1	
Nombre de la tarea: Describir el sistema propuesto y creación del modelo de dominio.	
Tipo de tarea: Desarrollo	
Fecha de inicio: 12 de marzo de 2013	Fecha de fin: 14 de marzo de 2013
Programador responsable: Rosalba Rivera Peguero	

Descripción: En la creación del modelo de dominio se describirá el sistema ayudando a detectar las clases o tablas principales para la creación de la base de datos.

Tabla 34. TI_2 Diseñar base de datos.

Tarea de Ingeniería	
Número de la tarea: 2	
Nombre de la tarea: Diseñar base de datos.	
Tipo de tarea: Desarrollo	
Fecha de inicio: 15 de marzo de 2013	Fecha de fin: 18 de marzo de 2013
Programador responsable: Rosalba Rivera Peguero	
Descripción: Para el desarrollo de la base de datos se utilizará PostgreSQL como sistema gestor, se crearán todas las tablas donde se almacenarán los datos persistentes generados durante la utilización de la aplicación.	

Tabla 35. TI_3 Describir los requisitos funcionales del sistema.

Tarea de Ingeniería	
Número de la tarea: 3	
Nombre de la tarea: Describir los requisitos funcionales del sistema.	
Tipo de tarea: Desarrollo	
Fecha de inicio: 19 de marzo de 2013	Fecha de fin: 21 de marzo de 2013
Programador responsable: Edney Nápoles Mejías	
Descripción: Para describir los requisitos funcionales es necesario entrevistarse con el cliente y conocer que desea realizar en cada uno de los requisitos funcionales detectados, ayudando a la confección de las interfaces.	

Tabla 36. TI_4 Confeccionar historias de usuario.

Tarea de Ingeniería	
Número de la tarea: 4	
Nombre de la tarea: Confeccionar historias de usuario.	
Tipo de tarea: Desarrollo	
Fecha de inicio: 22 de marzo de 2013	Fecha de fin: 25 de marzo de 2013
Programador responsable: Rosalba Rivera Peguero	
Descripción: Conociendo la descripción de las funcionalidades se desarrollarán las historias de usuario, como parte de la descripción del sistema.	

Tabla 37. TI_5 Generar las clases modelo.

Tarea de Ingeniería	
Número de la tarea: 5	
Nombre de la tarea: Generar las clases modelo.	
Tipo de tarea: Desarrollo	
Fecha de inicio: 25 de marzo de 2013	Fecha de fin: 26 de marzo de 2013
Programador responsable: Edney Nápoles Mejías	
Descripción: Se generan las clases modelo, especificándole el servidor de bases de datos, el nombre de la base de datos, el puerto a usar, el usuario y la contraseña para la conexión.	

Tabla 38. TI_6 Diseñar las interfaces de usuario.

Tarea de Ingeniería	
Número de la tarea: 6	
Nombre de la tarea: Diseñar las interfaces de usuario.	
Tipo de tarea: Desarrollo	

Fecha de inicio: 27 de marzo de 2013	Fecha de fin: 30 de marzo de 2013
Programador responsable: Rosalba Rivera Peguero	
Descripción: Para diseñar las interfaces será necesario guiarse por cada una de las funcionalidades descritas en la entrevista con el cliente y las historias de usuario.	

Tabla 39. TI_7 Implementar las funcionalidades descritas en las historias de usuario.

Tarea de Ingeniería	
Número de la tarea: 7	
Nombre de la tarea: Implementar las funcionalidades descritas en las historias de usuario.	
Tipo de tarea: Desarrollo	
Fecha de inicio: 31 de marzo de 2013	Fecha de fin: 30 de abril de 2013
Programador responsable: Edney Nápoles Mejías	
Descripción: Se implementan en las clases controladoras los métodos descritos en las historias de usuario.	

Anexos 4.

Las pruebas de aceptación se realizaron para comprobar el correcto funcionamiento de los requisitos implementados.

Tabla 40. Prueba de aceptación PA_1 para historia de usuario HU_1.

Caso de prueba de aceptación	
Código: PA_1 HU_1	Historia de usuario: HU_1
Nombre: Autenticar usuario	
Descripción: Prueba para las funcionalidades de autenticar usuario y verificar datos del usuario.	
Condiciones de ejecución: El usuario debe autenticarse para poder observar si se cumplen las operaciones e incorporar datos incorrectos para la validación.	

Pasos de ejecución: Se intenta acceder al sitio con datos incorrectos y los datos correctos del usuario.
Resultados esperados: Se accede al sistema correctamente y en caso de incorporar datos erróneos muestra mensaje de error.
Evaluación de prueba: Satisfactoria

Tabla 41. Prueba de aceptación PA_2 para historia de usuario HU_2.

Caso de prueba de aceptación	
Código: PA_2 HU_2	Historia de usuario: HU_2
Nombre: Realizar encuesta de estilo aprendizaje	
Descripción: Prueba para verificar que la encuesta se realice correctamente y permita el registro del estudiante.	
Condiciones de ejecución: Para realizar la encuesta debe ser estudiante y estar autenticado en el sistema.	
Pasos de ejecución: Se intenta realizar la encuesta por un usuario que contiene sus datos válidos.	
Resultados esperados: La encuesta es realizada sin generar error alguno.	
Evaluación de prueba: Satisfactoria	

Tabla 42. Prueba de aceptación PA_3 para historia de usuario HU_3.

Caso de prueba de aceptación	
Código: PA_3 HU_3	Historia de usuario: HU_3
Nombre: Registrar estudiante	
Descripción: Prueba para verificar el registro del estudiante	
Condiciones de ejecución: El usuario debe estar autenticado para efectuar la operación.	

Pasos de ejecución: Cuando se verifiquen los datos del usuario y este sea estudiante podrá realizar la encuesta de estilo de aprendizaje, una vez terminada se guardan los datos del usuario y el resultado de la encuesta en la tabla estudiante culminando el registro.
Resultados esperados: Se realiza la operación correctamente sin generar error.
Evaluación de prueba: Satisfactoria

Tabla 43. Prueba de aceptación PA_4 para historia de usuario HU_4.

Caso de prueba de aceptación	
Código: PA_4 HU_1	Historia de usuario: HU_4
Nombre: Ver listado estudiantes	
Descripción: Prueba para la funcionalidad ver listado de estudiantes registrados en el sistema	
Condiciones de ejecución: El usuario debe estar autenticado y validado que sea profesor para realizar la funcionalidad.	
Pasos de ejecución: Se intenta realizar la operación por un usuario con los permisos	
Resultados esperados: Se muestran correctamente los datos de los estudiantes registrados en el sistema.	
Evaluación de prueba: Satisfactoria	

Tabla 44. Prueba de aceptación PA_5 para historia de usuario HU_5.

Caso de prueba de aceptación	
Código: PA_5 HU_5	Historia de usuario: HU_5
Nombre: Gestionar tarea de estudiantes	
Descripción: Prueba para la funcionalidad de adicionar y eliminar las tareas de los estudiantes.	
Condiciones de ejecución: El usuario debe estar autenticado y validado que sea profesor para realizar la	

funcionalidad.
Pasos de ejecución: Se intenta adicionar y eliminar una tarea.
Resultados esperados: Las tareas son adicionadas y eliminadas correctamente.
Evaluación de prueba: Satisfactoria

Tabla 45. Prueba de aceptación PA_6 para historia de usuario HU_6.

Caso de prueba de aceptación	
Código: PA_6 HU_1	Historia de usuario: HU_6
Nombre: Asignar materiales para resolver tareas	
Descripción: Prueba para la funcionalidad de asignar los materiales a los estudiantes con cierto estilo de aprendizaje.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea estudiante para realizar la operación.	
Pasos de ejecución: Se intenta realizar la operación seleccionando una tarea para resolverla y la habilidad específica de la tarea.	
Resultados esperados: Se muestran los materiales necesarios para la tarea según el estilo de aprendizaje del estudiante.	
Evaluación de prueba: Satisfactoria	

Tabla 46. Prueba de aceptación PA_7 para historia de usuario HU_7.

Caso de prueba de aceptación	
Código: PA_7 HU_7	Historia de usuario: HU_7
Nombre: Ver tareas	
Descripción: Prueba para la funcionalidad de ver el listado de tareas agregadas al sistema.	

Condiciones de ejecución: El usuario debe estar autenticado y validado que sea profesor o estudiante, ambos tienen formas diferentes de ver el listado de tareas.
Pasos de ejecución: Se selecciona la operación a realizar para mostrar las tareas adicionales en el sistema.
Resultados esperados: Se muestran correctamente las tareas.
Evaluación de prueba: Satisfactoria

Tabla 47. Prueba de aceptación PA_8 para historia de usuario HU_8.

Caso de prueba de aceptación	
Código: PA_8 HU_8	Historia de usuario: HU_8
Nombre: Subir tareas resueltas	
Descripción: Prueba para la funcionalidad de subir una tarea una vez resuelta.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea estudiante para realizar la operación.	
Pasos de ejecución: Se intenta resolver una tarea y se busca la opción para adjuntar el resultado.	
Resultados esperados: Se adjunta correctamente la tarea del estudiante.	
Evaluación de prueba: Satisfactoria	

Tabla 48. Prueba de aceptación PA_9 para historia de usuario HU_9.

Caso de prueba de aceptación	
Código: PA_9 HU_9	Historia de usuario: HU_9
Nombre: Evaluar tareas	
Descripción: Prueba para la funcionalidad evaluar tareas de los estudiantes.	

Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación.
Pasos de ejecución: Se intenta seleccionar el estudiante para darle la evaluación y se escoge la evaluación correcta.
Resultados esperados: Se evalúa el estudiante correctamente.
Evaluación de prueba: Satisfactoria

Tabla 49. Prueba de aceptación PA_10 para historia de usuario HU_10.

Caso de prueba de aceptación	
Código: PA_10 HU_10	Historia de usuario: HU_10
Nombre: Mostrar evaluación de tareas	
Descripción: Prueba para la funcionalidad mostrar evaluación de los estudiantes.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación seleccionada.	
Pasos de ejecución: Se selecciona un estudiante del listado, la tarea que el profesor revisó y se selecciona ver evaluación.	
Resultados esperados: Se muestra la evaluación correctamente.	
Evaluación de prueba: Satisfactoria	

Tabla 50. Prueba de aceptación PA_11 para historia de usuario HU_11.

Caso de prueba de aceptación	
Código: PA_11 HU_11	Historia de usuario: HU_11
Nombre: Gestionar material educativo	
Descripción: Prueba para la funcionalidad de adicionar y eliminar un material educativo.	

Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar las operaciones seleccionadas.
Pasos de ejecución: Se intenta adicionar y eliminar un material educativo.
Resultados esperados: Se agregan y eliminan los materiales educativos correctamente.
Evaluación de prueba: Satisfactoria

Tabla 51. Prueba de aceptación PA_12 para historia de usuario HU_12.

Caso de prueba de aceptación	
Código: PA_12 HU_12	Historia de usuario: HU_12
Nombre: Ver material educativo	
Descripción: Prueba para la funcionalidad mostrar material educativo.	
Condiciones de ejecución: No es necesario estar autenticado para realizar la operación.	
Pasos de ejecución: Se intenta mostrar los materiales educativos existentes en el sistema.	
Resultados esperados: Se muestran correctamente los materiales dando la opción de abrirlo o descargarlo.	
Evaluación de prueba: Satisfactoria	

Tabla 52. Prueba de aceptación PA_13 para historia de usuario HU_13.

Caso de prueba de aceptación	
Código: PA_13 HU_13	Historia de usuario: HU_13
Nombre: Realizar encuesta de satisfacción	
Descripción: Prueba para la funcionalidad realizar encuesta de satisfacción de los estudiantes con el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea estudiante para realizar la	

operación.
Pasos de ejecución: Se intenta realizar la encuesta de satisfacción por el usuario autenticado.
Resultados esperados: Se realiza la operación correctamente guardando su resultado.
Evaluación de prueba: Satisfactoria

Tabla 53. Prueba de aceptación PA_14 para historia de usuario HU_14.

Caso de prueba de aceptación	
Código: PA_14 HU_14	Historia de usuario: HU_14
Nombre: Mostrar resultado encuesta de satisfacción	
Descripción: Prueba para la funcionalidad mostrar resultado de la encuesta de satisfacción.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación.	
Pasos de ejecución: Se intenta mostrar el resultado de la encuesta para conocer lo que piensa el estudiante respecto al servicio que presta el sistema.	
Resultados esperados: Se muestra la información recogida correctamente.	
Evaluación de prueba: Satisfactoria	

Tabla 54. Prueba de aceptación PA_15 para historia de usuario HU_15.

Caso de prueba de aceptación	
Código: PA_15 HU_15	Historia de usuario: HU_15
Nombre: Gestionar asignatura	
Descripción: Prueba para la funcionalidad de adicionar y eliminar asignatura.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la	

operación.
Pasos de ejecución: Se intenta adicionar y eliminar una asignatura en el sistema.
Resultados esperados: Se adiciona y elimina correctamente guardando la información en la base de datos.
Evaluación de prueba: Satisfactoria

Tabla 55. Prueba de aceptación PA_16 para historia de usuario HU_16.

Caso de prueba de aceptación	
Código: PA_16 HU_16	Historia de usuario: HU_16
Nombre: Gestionar tema	
Descripción: Prueba para la funcionalidad de adicionar y eliminar tema.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación.	
Pasos de ejecución: Se intenta adicionar y eliminar un tema en el sistema.	
Resultados esperados: Se adiciona y elimina correctamente guardando la información en la base de datos.	
Evaluación de prueba: Satisfactoria	

Tabla 56. Prueba de aceptación PA_17 para historia de usuario HU_17.

Caso de prueba de aceptación	
Código: PA_17 HU_17	Historia de usuario: HU_17
Nombre: Gestionar habilidades	
Descripción: Prueba para la funcionalidad de adicionar y eliminar las habilidades de una tarea.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la	

operación.
Pasos de ejecución: Se intenta adicionar y eliminar una habilidad en el sistema.
Resultados esperados: Se adiciona y elimina correctamente guardando la información en la base de datos.
Evaluación de prueba: Satisfactoria

Tabla 57. Prueba de aceptación PA_18 para historia de usuario HU_18.

Caso de prueba de aceptación	
Código: PA_18 HU_18	Historia de usuario: HU_18
Nombre: Gestionar habilidades específicas	
Descripción: Prueba para la funcionalidad de adicionar y eliminar las habilidades específicas de una tarea y de cierta habilidad.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación.	
Pasos de ejecución: Se intenta adicionar y eliminar una habilidad específica en el sistema.	
Resultados esperados: Se adiciona y elimina correctamente guardando la información en la base de datos.	
Evaluación de prueba: Satisfactoria	

Tabla 58. Prueba de aceptación PA_19 para historia de usuario HU_19.

Caso de prueba de aceptación	
Código: PA_19 HU_19	Historia de usuario: HU_19
Nombre: Gestionar material para las habilidades específicas	
Descripción: Prueba para la funcionalidad de adicionar y eliminar materiales para resolver las tareas según la habilidad específica.	

Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación.
Pasos de ejecución: Se intenta adicionar y eliminar un material específico en el sistema.
Resultados esperados: Se adiciona y elimina correctamente guardando la información en la base de datos.
Evaluación de prueba: Satisfactoria

Tabla 59. Prueba de aceptación PA_20 para historia de usuario HU_20.

Caso de prueba de aceptación	
Código: PA_20 HU_20	Historia de usuario: HU_20
Nombre: Relacionar habilidad tarea	
Descripción: Prueba para la funcionalidad de relacionar una habilidad a las tareas.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación.	
Pasos de ejecución: Se selecciona la habilidad y la tarea a relacionar.	
Resultados esperados: Se relacionan correctamente.	
Evaluación de prueba: Satisfactoria	

Tabla 60. Prueba de aceptación PA_21 para historia de usuario HU_21.

Caso de prueba de aceptación	
Código: PA_21 HU_21	Historia de usuario: HU_21
Nombre: Mostrar tareas resueltas adjuntadas	
Descripción: Prueba para la funcionalidad mostrar las tareas adjuntadas por los estudiantes para que los profesores puedan revisarla.	

Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación.
Pasos de ejecución: Se intenta mostrar las tareas adjuntadas por los estudiantes.
Resultados esperados: Se muestran correctamente las tareas dando la opción de descargarla para su revisión.
Evaluación de prueba: Satisfactoria

Tabla 61. Prueba de aceptación PA_22 para historia de usuario HU_22.

Caso de prueba de aceptación	
Código: PA_22 HU_22	Historia de usuario: HU_22
Nombre: Eliminar estudiante	
Descripción: Prueba para la funcionalidad eliminar estudiantes del sistema.	
Condiciones de ejecución: El usuario debe estar autenticado y validado sea profesor para realizar la operación.	
Pasos de ejecución: Se selecciona la operación eliminar estudiantes y se escoge el estudiante a eliminar.	
Resultados esperados: Se eliminan correctamente los estudiantes del sistema y todos sus datos.	
Evaluación de prueba: Satisfactoria	