



Universidad de las Ciencias
Informáticas

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

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

Sistema informático para confeccionar instrumentos de
evaluación en la Facultad 2 de la Universidad de las
Ciencias Informáticas.

Autores: Félix Ramón Quintero Mora
Reinaldo Enrique Merino Hernández

Tutor: Ing. Yusniel Ávila Malagón
Co-Tutor: Ing. Alejandro Arias Naranjo

La Habana, junio de 2014

“Año 56 de la Revolución”

“La educación es el arma más potente para cambiar el mundo.”

Nelson Mandela (1918-2013)



Dedicatoria de Félix

Al tesoro más grande de mi vida, mi razón de ser, mi mejor amiga, mi todo, a ti mamita te dedico este trabajo, fruto de tantos sacrificios y de una distancia endemoniada. Gracias por apoyarme siempre, por todo tu amor, comprensión y por ser la mejor madre del mundo.

A mis dos tatas, Darito y Chuco sin ustedes no sé qué sería de mí. Los amo con todo mi corazón, son los mejores hermanos que alguien pudiera tener, gracias por estar siempre en mi vida y por hacerme saber que siempre voy a contar con su cariño.

A mi bruji Laura por ser mi única en el mundo, por amarme tanto, por darme el aliento cuando más lo necesitaba, por acompañarme en mis locuras, por creer en mí cuando yo mismo no lo hacía, por ser el amor de mi vida y mi psicóloga particular. Te amo con todo mi ser.

Dedicatoria de Reinaldo

A todos los que aparecen en los agradecimientos, pero también al hombre que hizo posible que esta Universidad hoy por hoy exista, a Fidel.

Agradecimientos de Félix

Quiero agradecer de forma muy especial a mi papá, siempre he querido ser un orgullo para ti y aunque no sea un gran doctor, creo que soy un gran hombre y en parte te lo debo a ti.

A tía Mary por todo el amor, el apoyo, por ser la segunda mamá mía y de mis hermanos. Por ser una guía y una luz a seguir. Je quiero mucho. Ya casi vienen los zapatos.

A mi abuelita Angelina quien me crió con tanto amor y disciplina, por siempre estar preocupada por mis resultados. Je adoro y lo haré siempre.

A tía Rosy por ser tan maravillosa y estar pendiente de todos nuestros problemas, gracias por siempre estar ahí.

Al resto de mi familia, a todos los primos, los que están cerca y los que no. De todos he aprendido mucho, gracias por formar todos juntos una familia maravillosa.

A Niurka por ser la mejor suegra del mundo y por todo el apoyo que me has dado.

A Ziomara por ser como una abuelita para mí, por tu cariño y tus consejos.

A José, Eli y Peque por ser mi nueva familia. José gracias por ser mi amigo, por todo el cariño, esta es nuestra tesis.

A José Nandin, mi hermano del pre y de toda la vida, una de las personas más brillantes que he conocido, siempre estás en mi corazón.

A todos los nuevos amigos que hice en la UCI, a Yong, Carlos F., El malva, Marcel, Carlos C., Jorge Martínez, Jorgito, Argudín y Yasmani.

A todo el pikete del futsal más malo del mundo, nunca hicimos nada pero nos divertimos siempre.

A todo el grupo fantasma 7105, es difícil olvidar el poco tiempo que estuvimos juntos.

A Roxana por todo el sacrificio, por ser la mejor profesora que tuve, por ser una gran amiga, por casi embarcarme con esta tesis, sin su ayuda no lo hubiera logrado.

A Maidelis por aparecer en el momento adecuado como ese ángel de la guardia y echarse todo el esfuerzo final de esta tesis encima. Muchísimas gracias, siempre voy a estar en deuda con usted.

A Ávila por nunca abandonarnos en este camino, por estar en los momentos precisos.

A todos los que han formado parte de mi vida y que se me pueden quedar, es difícil recordarlos a todos al momento de redactar por los sentimientos que afloran.

Por último y no por eso menos importante agradecer a mi compañero de tesis, a ti Rey te deseo lo mejor de este mundo, siempre podrás contar conmigo.

Agradecimientos de Reinaldo

Quiero comenzar a agradecer de manera general a todos los que han tenido que ver y han influido en hecho de que me pueda graduar como Ingeniero Informático. Agradezco a todos mis compañeros, desde los que están aún hoy conmigo en el aula hasta los que se fueron desde el mismo primer año, especialmente a Marcel y Carlos Ernesto, dos de aquella época que ya no se encuentran hoy aquí conmigo.

Quiero agradecer a todos los profesores con los que compartí aula, quienes más que profesores fueron amigos y familia en las buenas y en las malas.

Agradecer a mis compañeros de aula por ser quienes son y por siempre estar ahí.

Especial agradecimiento a Dayana, gran amiga en quien siempre he podido confiar a pesar de no callarse nada. A Oscar Leandro por ser fiel compañía durante estos años de estudios. A Keilor especial agradecimiento por todo su apoyo y compañía en todos los momentos y por ser esa

persona que siempre estaba ahí a la hora que se necesitara y que siempre, sin importar lo mala que fuese la situación, se las arreglaba para hacerme reír.

Quiero agradecerle a José Miguel por brindarme su amistad y su casa en los momentos en los que más necesité compañía y apoyo, a Damaris por estar ahí conmigo durante gran parte de la carrera a mi lado aguantando mis cosas. A Frank Ernesto por ser como un hermano incondicional, no tengo palabras para describirlo.

A mi compañero de tesis, Félix, por estar conmigo durante estos 5 años y ser otro de los hermanos que hice aquí en la UCI. Gracias a él esta tesis sale hoy.

Agradezco a la profe Roxana, que ha sido mi mamá sustituta en estos años, siempre aconsejándome y llevándome por el buen camino, aunque haya tenido que coger una sombrilla en la mano en más de una ocasión y no exactamente para darme sombra.

Agradecer a la gente bonita con la que he convivido, a Carlos, Yaiselis, Osvaldo y Darling, todos compartiendo conmigo el día a día de este último año.

Especial agradecimiento a mis padres por todo el apoyo y amor que me han dado, por todas sus preocupaciones y por todas las carreras que han dado para hacer más pasajera mi estancia aquí.

Agradecer también a Abel, quien ha estado a mi lado día a día todo este año, apoyándome en todos los sentidos en este proceso complejo que es la realización de una tesis y acompañándome en mi vida diaria.

RESUMEN

La presente investigación tiene como objetivo desarrollar un sistema informático que permita optimizar el proceso de confección de instrumentos evaluativos en la Facultad 2 de la Universidad de Ciencias Informáticas.

La propuesta es una aplicación de escritorio que permite la generación de instrumentos evaluativos de diferentes asignaturas, éstos son generados de forma dinámica o selectiva. Las preguntas contenidas en la base de datos se presentan por asignaturas, por temas, por objetivos y por niveles de complejidad. Cada nivel corresponde a un tipo de pregunta que se identifica por las habilidades que se pretende lograr, un primer nivel de complejidad (bajo) con preguntas de carácter reproductivo; un segundo nivel (medio) con preguntas de carácter aplicativo y un tercer nivel (alto) con preguntas creativas que son las que requieren de un nivel intelectual más alto, estos niveles son asignados por los jefes de las asignaturas.

El desarrollo de dicho sistema está guiado por un enfoque ágil utilizando la metodología Programación Extrema (XP), se codificó con el lenguaje de programación C# sobre el entorno integrado de desarrollo Microsoft Visual Studio 2010 y como gestor de base de datos Microsoft Server 2008.

Palabras clave: Instrumentos evaluativos, niveles de complejidad, sistema informático

<i>INTRODUCCIÓN</i>	1
<i>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN</i>	6
1.1. Tecnología Educativa para la evaluación	6
1.2. Proceso docente educativo	7
1.3. Evaluación del aprendizaje en la educación superior	8
1.4. Funciones de la evaluación.....	9
1.5. Cualidades de los instrumentos de evaluación.....	10
1.6. Sistemas existentes para confeccionar instrumentos evaluativos en el ámbito internacional	11
1.7. Sistemas existentes para confeccionar instrumentos evaluativos en el ámbito nacional	14
1.8. Análisis de los sistemas estudiados para confeccionar instrumentos evaluativos.....	15
1.9. Herramientas, técnicas y tecnologías a utilizar	17
1.9.1. Algoritmos criptográficos	17
1.9.2. Metodología de desarrollo	19
1.9.3. Lenguaje de programación.....	23
1.9.4. Entornos Integrados de Desarrollo	24
1.9.5. Gestor de Bases de Datos	25
1.9.6. Lenguaje de modelado.....	25
1.9.7. Herramientas para el diseño de la aplicación	26
1.10. Conclusiones parciales	27
<i>CAPÍTULO 2. EXPLORACIÓN Y PLANIFICACIÓN</i>	28
2.1. Características de la propuesta de solución.....	28
2.2. Usuarios relacionados con el sistema	29
2.3. Exploración	29
2.3.1. Historias de Usuarios	30
2.4. Planificación.....	35
2.4.1. Estimación de esfuerzos por HU	35
2.5. Plan de iteraciones	36

2.5.1. Plan de duración de iteraciones	37
2.5.2. Plan de entregas	37
2.6. Conclusiones parciales	38
<i>CAPÍTULO 3. ITERACIONES</i>	39
3.1. Diseño del sistema	39
3.2. Diseño de la Base de Datos.....	40
3.3. Patrones de Arquitectura	41
3.4. Patrones de Diseño	42
3.5. Estilos y estándares de codificación	43
3.6. Fase de Iteraciones	44
3.7. Pruebas	51
3.8. Conclusiones parciales	60
<i>CONCLUSIONES GENERALES</i>	61
<i>RECOMENDACIONES</i>	62
<i>REFERENCIA BIBLIOGRÁFICA</i>	63
<i>BIBLIOGRAFÍA</i>	67
<i>ANEXOS</i>	71
<i>GLOSARIO DE TÉRMINOS</i>	75

<i>Tabla 1. Comparación entre los sistemas estudiados</i>	15
<i>Tabla 2 Ranking de “agilidad” (18)</i>	21
<i>Tabla 3. Usuarios relacionados con el sistema</i>	29
<i>Tabla 4. HU Autenticar Usuario</i>	31
<i>Tabla 5. HU Gestionar Pregunta</i>	31
<i>Tabla 6. HU Gestionar Exámenes</i>	32
<i>Tabla 7. HU Exportar exámenes a PDF</i>	32
<i>Tabla 8. HU Asignar asignatura</i>	32
<i>Tabla 9. HU Asignar complejidad</i>	33
<i>Tabla 10. HU Gestionar usuario</i>	33
<i>Tabla 11. HU Gestionar asignatura</i>	33
<i>Tabla 12. HU Gestionar Tema</i>	34
<i>Tabla 13. HU Gestionar Objetivos</i>	34
<i>Tabla 14. Seguridad</i>	34
<i>Tabla 15. Usabilidad</i>	35
<i>Tabla 16. Condiciones tecnológicas</i>	35
<i>Tabla 17. Estimación de esfuerzos por HU</i>	36
<i>Tabla 18. Plan de duración de iteraciones</i>	37
<i>Tabla 19. Plan de entregas</i>	38
<i>Tabla 20. Modelo de las tarjetas CRC</i>	39
<i>Tabla 21. Tarjeta CRC: Usuario</i>	40
<i>Tabla 22. Tarjeta CRC: Autenticar</i>	40
<i>Tabla 23. Tarjeta CRC: AES</i>	40
<i>Tabla 24. Tarjeta CRC: PDF</i>	40
<i>Tabla 25. Tarjeta CRC: Conexión</i>	40

<i>Tabla 26. Tarea de Ingeniería #1</i>	44
<i>Tabla 27. Tarea de Ingeniería #2</i>	45
<i>Tabla 28. Tarea de Ingeniería #3</i>	45
<i>Tabla 29. Tarea de Ingeniería #4</i>	45
<i>Tabla 30. Tarea de Ingeniería #5</i>	45
<i>Tabla 31. Tarea de Ingeniería #6</i>	46
<i>Tabla 32. Tarea de Ingeniería #7</i>	46
<i>Tabla 33. Tarea de Ingeniería #8</i>	46
<i>Tabla 34. Tarea de Ingeniería #9</i>	46
<i>Tabla 35. Tarea de Ingeniería #10</i>	47
<i>Tabla 36. Tarea de Ingeniería #11</i>	47
<i>Tabla 37. Tarea de Ingeniería #12</i>	47
<i>Tabla 38. Tarea de Ingeniería #13</i>	48
<i>Tabla 39. Tarea de Ingeniería #14</i>	48
<i>Tabla 40. Tarea de Ingeniería #15</i>	48
<i>Tabla 41. Tarea de Ingeniería #16</i>	49
<i>Tabla 42. Tarea de Ingeniería #17</i>	49
<i>Tabla 43. Tarea de Ingeniería #18</i>	49
<i>Tabla 44. Tarea de Ingeniería #19</i>	49
<i>Tabla 45. Tarea de Ingeniería #20</i>	50
<i>Tabla 46. Tarea de Ingeniería #21</i>	50
<i>Tabla 47. Tarea de Ingeniería #22</i>	50
<i>Tabla 48. Tarea de Ingeniería #23</i>	50
<i>Tabla 49. Tarea de Ingeniería #24</i>	51
<i>Tabla 50. Tarea de Ingeniería #25</i>	51

<i>Tabla 51. Tarea de Ingeniería #26.....</i>	<i>51</i>
<i>Tabla 52. Caso de prueba de aceptación #1</i>	<i>53</i>
<i>Tabla 53. Caso de prueba de aceptación #2</i>	<i>53</i>
<i>Tabla 54. Caso de prueba de aceptación #3</i>	<i>54</i>
<i>Tabla 55. Caso de prueba de aceptación #4</i>	<i>54</i>
<i>Tabla 56. Caso de prueba de aceptación #5</i>	<i>55</i>
<i>Tabla 57. Caso de prueba de aceptación #6</i>	<i>55</i>
<i>Tabla 58. Caso de prueba de aceptación #7</i>	<i>56</i>
<i>Tabla 59. Caso de prueba de aceptación #8</i>	<i>56</i>
<i>Tabla 60. Caso de prueba de aceptación #9</i>	<i>57</i>
<i>Tabla 61. Caso de prueba de aceptación #10.....</i>	<i>57</i>
<i>Tabla 62. Caso de prueba de aceptación #11</i>	<i>57</i>
<i>Tabla 63. Caso de prueba de aceptación #12</i>	<i>58</i>
<i>Tabla 64. Caso de prueba de aceptación #13.....</i>	<i>58</i>
<i>Tabla 65. Caso de prueba de aceptación #14.....</i>	<i>58</i>
<i>Tabla 66. Caso de prueba de aceptación #15.....</i>	<i>59</i>
<i>Tabla 67. Caso de prueba de aceptación #16.....</i>	<i>59</i>
<i>Tabla 68. Caso de prueba de aceptación #17.....</i>	<i>59</i>
<i>Tabla 69. Cantidad de No Conformidades detectadas por iteración.....</i>	<i>60</i>
<i>Tabla 70. Caso de prueba de aceptación #23.....</i>	<i>71</i>
<i>Tabla 71. Caso de prueba de aceptación #24.....</i>	<i>72</i>
<i>Tabla 72. Caso de prueba de aceptación #25.....</i>	<i>72</i>
<i>Tabla 73. Caso de prueba de aceptación #26.....</i>	<i>72</i>
<i>Tabla 74. Caso de prueba de aceptación #18.....</i>	<i>73</i>
<i>Tabla 75. Caso de prueba de aceptación #19.....</i>	<i>73</i>

<i>Tabla 76. Caso de prueba de aceptación #20</i>	74
<i>Tabla 77. Caso de prueba de aceptación #21</i>	74
<i>Tabla 78. Caso de prueba de aceptación #22</i>	74

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) han producido importantes transformaciones en la sociedad, éstas marcan la característica fundamental que distingue el momento histórico actual. Su uso intensivo ha elevado el desarrollo científico - técnico, involucrando a las más diversas esferas de la vida, como la salud, el deporte, la cultura y la educación.

La incorporación de las TIC en el ámbito de la educación ha evolucionado a lo largo de estos últimos años, tanto que la utilización de estas tecnologías en el aula ha pasado de ser una posibilidad, a erigirse como una necesidad y una herramienta de trabajo básica para el profesorado y el alumnado. (1)

Como resultado de las aplicaciones de diferentes concepciones y teorías educativas para la resolución de problemas y situaciones referidas a la enseñanza y el aprendizaje, apoyadas en las TIC, surge el concepto Tecnología Educativa, que se resume en, "el modo sistemático de concebir, aplicar y evaluar el conjunto de procesos de enseñanza y aprendizaje, teniendo en cuenta a la vez los recursos técnicos y humanos y las interacciones entre ellos, como forma de obtener una más efectiva educación." (2)

Según el informe de la República de Cuba en la Cumbre Mundial sobre la Sociedad de la Información celebrada en Ginebra en diciembre de 2003, los objetivos fundamentales para transformar los sistemas educativos nacionales, en función del desarrollo integral de sus ciudadanos e insertarse en la sociedad de la información, son: "elevar la calidad de la educación; garantizar la necesaria preparación en las TIC de los recursos humanos; instrumentar (orientar y organizar) un proceso de educación continua; y ampliar la cultura general de la población sobre estas tecnologías." (3)

Bajo el contexto actual del país es necesario tener en cuenta la actualización que sufre el sistema social cubano, guiado por los lineamientos de la política económica y social del Partido y la Revolución, durante el transcurso de la investigación se evidencia que existen lineamientos con los cuales se puede contribuir de forma directa:

Lineamiento 145. Continuar avanzando en la elevación de la calidad y rigor del proceso docente educativo, jerarquizar la superación permanente, el enaltecimiento y atención del personal docente, y el papel de la familia en la educación de niños y jóvenes. Lograr una mejor utilización y aprovechamiento de la fuerza de trabajo y de las capacidades existentes.

Lineamiento 147. Fortalecer el papel del profesor frente al alumno y lograr que los equipos y medios audiovisuales sean un complemento de la labor educativa del docente y garantizar el uso racional de los mismos.

La Universidad de las Ciencias Informáticas (UCI), uno de los centros rectores de la educación superior en el país, tiene como misión la formación de profesionales comprometidos con su Patria y altamente calificados en su rama. La formación de los profesionales se desarrolla a través del proceso docente educativo, el cual está dirigido a la formación social de las nuevas generaciones. Uno de sus eslabones esenciales es la evaluación del aprendizaje, que tiene como propósito comprobar el grado de cumplimiento de los objetivos formulados en los planes y programas de estudio, mediante la valoración de los conocimientos y habilidades que los estudiantes van adquiriendo durante la carrera.

La evaluación del aprendizaje en la UCI se basa en la Resolución 210 del 2007 del Ministerio de Educación Superior y se estructura en tres tipos fundamentales: evaluación frecuente, parcial y final. Dentro de las evaluaciones frecuentes se encuentran las preguntas escritas, orales y las discusiones grupales; en el grupo de las evaluaciones parciales aparecen la prueba parcial, el trabajo extraclase y el encuentro comprobatorio; y como evaluaciones finales los tipos fundamentales son: el examen final, la defensa del trabajo de curso y la evaluación final de la práctica laboral.

De las evaluaciones mencionadas anteriormente, los exámenes parciales y finales, para ser aplicadas necesitan la confección de un instrumento de evaluación, que involucra al colectivo de profesores de la asignatura a evaluar. El proceso de confección de los instrumentos se realiza de forma manual: se reúnen los profesores de la asignatura que se va a evaluar, cada uno realiza una o varias propuestas de interrogantes y se seleccionan las que más se ajustan a los objetivos, teniendo en cuenta un balance adecuado en la complejidad de las mismas. Las pruebas parciales se elaboran a nivel de Facultad, mientras que los exámenes finales a nivel de Departamentos Metodológicos Centrales.

En el Departamento de Programación de la Facultad 2 la elaboración de un instrumento de evaluación es un proceso complejo, pues el colectivo de las asignaturas que se imparten está compuesto por profesores con poca o ninguna experiencia pedagógica. El claustro está conformado por 34 profesores, de ellos 14 están categorizados como asistentes y 20 como instructores, lo que provoca que en ocasiones los instrumentos no cumplan con los requisitos metodológico definidos para los mismos, teniendo en cuenta la

función que debe cumplir la evaluación, el equilibrio en los niveles de complejidad de acuerdo con los objetivos y el nivel de asimilación.

La poca experiencia del claustro provoca también que en las reuniones para confeccionar los instrumentos no siempre se garantice el equilibrio entre las propuestas de los profesores, pues estos no tienen en cuenta los aspectos mencionados anteriormente, lo cual provoca que se incurra en pérdida de tiempo durante dicho proceso para seleccionar las preguntas definitivas que van a estar en el temario a aplicar.

No existe en el departamento un repositorio digital con exámenes aplicados en años anteriores que sirva de referencia para el trabajo de los docentes y como guía para no repetir algunas preguntas ya evaluadas, los temarios archivados hasta el momento se encuentran en formato duro y se hace difícil el acceso a estos.

Por lo expuesto anteriormente se plantea como **problema de la investigación**: ¿Cómo contribuir a optimizar el proceso de confección de instrumentos evaluativos en la Facultad 2 de la Universidad de Ciencias Informáticas?

A partir del problema a resolver planteado se puede inferir que el **objeto de estudio** lo constituye, las tecnologías educativas para la evaluación y como **campo de acción**, los sistemas para la confección de instrumentos evaluativos en la Facultad 2 de la Universidad de las Ciencias Informáticas.

El presente trabajo tiene como **objetivo general**: desarrollar un sistema informático para la confección de instrumentos evaluativos en la Facultad 2 de la Universidad de las Ciencias Informáticas que logre optimizar este proceso de acuerdo a los requerimientos metodológicos establecidos.

Defendiendo como idea que si se desarrolla un sistema informático que garantice la correspondencia entre la complejidad de las preguntas propuestas para evaluar un tema de una asignatura y sus objetivos, avalado por un criterio de experto, formando parte de un repositorio se logrará optimizar el proceso de elaboración de instrumentos evaluativos en la Facultad 2.

En correspondencia con el objetivo propuesto se trazan las siguientes **tareas de investigación**:

- Definición de los aspectos teóricos referentes a la elaboración de instrumentos evaluativos dentro del proceso docente educativo.

- Caracterización de sistemas existentes relacionados con la elaboración de instrumentos evaluativos, para identificar elementos que se ajusten al problema a resolver.
- Selección de la metodología de *software* para el desarrollo del sistema.
- Asimilación de las tecnologías y herramientas a utilizar en el proceso de desarrollo del sistema.
- Selección del algoritmo criptográfico para garantizar la seguridad de la base de datos del sistema.
- Definición de los requisitos que debe cumplir la propuesta de solución a desarrollar.
- Realización de pruebas de *software* al sistema para garantizar la calidad del mismo.

Resultados esperados:

Un sistema informático para la confección de instrumentos evaluativos en la Facultad 2 de la Universidad de las Ciencias Informáticas que optimice este proceso, estableciendo un balance entre la complejidad de las preguntas propuestas para evaluar un tema de una asignatura y sus objetivos, avalado por un criterio de experto. Se podrá contar con un repositorio digital de exámenes aplicados en años anteriores que sirva de referencia para el trabajo de los docentes y favorezca el trabajo colaborativo entre profesores.

Métodos de investigación

El empleo de los métodos tanto teóricos como empíricos utilizados en el discurso investigativo están sustentados en el método de investigación general **dialéctico - materialista**. En calidad de métodos teóricos se emplearon los métodos analítico - sintético e inductivo – deductivo y los métodos empíricos utilizados fueron la observación y la entrevista.

El método **analítico - sintético** permitió realizar un estudio bibliográfico de la teoría existente alrededor de las Tecnologías Educativas para la evaluación. A partir del mismo, se determinaron las características que tendría la propuesta de solución, se definieron cuáles serían las tecnologías y herramientas más adecuadas para el desarrollo de la propuesta de solución.

El método **inductivo - deductivo** permitió la precisión de los análisis teóricos y metodológicos concebidos en el objeto de estudio de la investigación, así como percibió lo más general y lo particular con relación al componente evaluación en el proceso docente educativo en la Universidad de las Ciencias Informáticas. Este método facilitó inducir conclusiones generales del análisis del objeto de estudio.

Mediante la **observación** del proceso de confección de instrumentos de evaluación en el Departamento de Programación en la Facultad 2 de la Universidad de las Ciencias Informáticas, se comprobó que la

elaboración de los mismos, al ser realizada de forma manual por un claustro joven de poca experiencia pedagógica presenta una serie de problemas relacionados a la gestión del tiempo y a la calidad en el desarrollo de los mismos, por lo que sería provechoso desarrollar una herramienta informática que contribuya a solventar estos problemas.

La **entrevista** es una conversación planificada entre el investigador y el entrevistado para obtener información sobre el problema a resolver, en la investigación se utilizó la entrevista no estructurada, cuyo objetivo es obtener criterios de expertos a través de especialistas en el tema tratado. Se entrevistó al jefe de Departamento de Programación de la Facultad 2 con el objetivo de conocer las características del claustro y a profesores del Departamento para determinar los problemas que presentan para confeccionar los instrumentos evaluativos.

Para una mejor comprensión de la investigación, cuyo diseño metodológico ha sido descrito, se decidió definir una estructura capitular que aporte cierto grado de organización y facilite la comprensión de la presente investigación. Los capítulos que la conforman, son los siguientes:

Capítulo 1: Fundamentación Teórica de la investigación: El capítulo aborda los aspectos teóricos que dan sustento a la investigación. Se incluye el análisis de los sistemas que permiten la confección de instrumentos evaluativos en el ámbito internacional y nacional, además la caracterización de las tecnologías, herramientas y técnicas a usar en el desarrollo de la propuesta de solución.

Capítulo 2: Exploración y Planificación: El capítulo está dedicado a la propuesta del sistema a desarrollar atendiendo a la metodología seleccionada. En la fase Exploración los clientes definen lo que necesitan mediante la redacción de las Historias de Usuario y el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Durante la fase Planificación el cliente establece la prioridad de cada Historia de Usuario y los programadores realizan una estimación del esfuerzo para implementarlas, se realiza el Plan de Iteraciones y el Plan de Entregas del producto.

Capítulo 3: Iteraciones: El capítulo está dedicado a la tercera fase de la metodología seleccionada definida para la investigación. Se presenta el diseño del sistema y se muestran las tareas de ingeniería necesarias para llevar a cabo el proceso de desarrollo. Se definen las estrategias de pruebas a seguir y se generan los casos de pruebas para realizar las validaciones junto al cliente.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

En el presente capítulo se definen elementos asociados al objeto de estudio de la investigación. Se hace un estudio de sistemas relacionados con el proceso de elaboración de instrumentos evaluativos a nivel internacional y nacional. Se analizan las herramientas, tecnologías, metodologías de desarrollo y los lenguajes de programación a utilizar en la realización de la aplicación, para darle cumplimiento al objetivo general de la investigación.

1.1. Tecnología Educativa para la evaluación

La evolución de la Tecnología Educativa (TE), como disciplina nació en Estados Unidos de América en la década de los 50 del siglo pasado, ha dado lugar a diferentes enfoques o tendencias que se han conocido como enseñanza audiovisual, enseñanza programada, tecnología instruccional, diseño curricular o tecnología crítica de la enseñanza. (4)

Se entiende por Tecnología Educativa al acercamiento científico basado en la teoría de sistemas que proporciona al educador las herramientas de planificación y desarrollo, así como la tecnología. Busca mejorar los procesos de enseñanza y de aprendizaje a través del logro de los objetivos educativos y la efectividad y el significado del aprendizaje. (5)

Entre las ventajas del uso de la Tecnología Educativa destacan no sólo las herramientas que pueden utilizarse, pues también influyen en los tres saberes que maneja el nuevo modelo constructivista de la educación: saber ser, saber saber y saber hacer; ya que favorecen una mayor autonomía en la calidad del conocimiento adquirido por los estudiantes a través del desarrollo de trabajos colaborativos que con la ayuda y mediación del asesor mejoren la capacidad de pensamiento de los alumnos permitiéndoles realizar análisis y reflexiones críticas. (6)

En la presente investigación se asocia la Tecnología Educativa con la confección de instrumentos de evaluación dentro del proceso docente educativo. Para entender el correcto funcionamiento de los sistemas tecnológicos que permiten confeccionar este tipo de instrumentos es necesario dominar los conceptos y regulaciones fundamentales referentes a la evaluación dentro del proceso docente educativo.

1.2. Proceso docente educativo

El proceso docente educativo es el proceso formativo escolar que del modo más sistémico se dirige a la formación social de las nuevas generaciones y en él, el estudiante se instruye, desarrolla y educa. Al analizarlo se puede apreciar, mediante la observación inmediata del mismo, la actividad del estudiante para instruirse: el aprendizaje. También se puede apreciar la actividad del profesor que guía ese aprendizaje; denominada enseñanza. (7)

Un análisis de esta observación puede llevar a la conclusión de que los componentes del proceso son el aprendizaje y la enseñanza. Sin embargo, un estudio más profundo del proceso docente educativo informa sobre otros componentes esenciales. La sociedad crea las instituciones docentes con el fin de resolver un problema de enorme trascendencia, este se denomina encargo social y consiste en la necesidad de preparar a los ciudadanos de esa sociedad, tanto en su pensamiento, como en sus sentimientos, junto con la preparación inmediata para su actividad laboral, en correspondencia con los valores más importantes de la misma. (7)

El **problema** es la situación que presenta un objeto y que genera en alguien una necesidad. Así pues, el encargo social es un problema, porque en este se concreta la necesidad que tiene la sociedad de preparar a sus ciudadanos con determinada formación, habilidades y valores para actuar en un contexto social en una época dada. (7)

El **objeto** es la parte de la realidad portador del problema. Es decir, es un aspecto del proceso productivo o de servicio, en el cual se manifiesta la necesidad de preparar a profesionales para que participen en la solución del problema, que se resuelve inmerso en el proceso de formación del ciudadano. El problema se vincula también con otro importante componente del proceso docente educativo: el **objetivo**. El objetivo es la aspiración que se pretende lograr en la formación de los ciudadanos del país y en particular de las nuevas generaciones, para resolver el problema. (7)

Para alcanzar ese objetivo el estudiante debe formar su pensamiento, cultivar sus facultades, mediante el dominio de una rama del saber que está presente en el objeto en que se manifiesta el problema, a esto le llamamos el **contenido** del aprendizaje. Cuando el estudiante se apropia del contenido, debe tener un cierto orden, una determinada secuencia, a este se le denomina **método**. (7)

La organización del tiempo, en correspondencia con el contenido a asimilar y el objetivo a alcanzar; así como la relación entre los estudiantes y el profesor, que viene dada por ejemplo, por la cantidad de estudiantes que estarán en el aula con el profesor en un momento determinado, se denomina **forma de enseñanza**. El proceso docente educativo se desarrolla con ayuda de algunos objetos, como son, el pizarrón, la tiza, los equipos de laboratorios, el televisor, la computadora, todo lo cual se denomina **medio de enseñanza**. (7)

El **resultado** es el componente que expresa las transformaciones que se lograron alcanzar en el escolar; es el producto que se obtiene del proceso. El vínculo de este con el objetivo durante un eslabón del proceso es llamado **evaluación**. La evaluación de la disciplina no se concibe a base de pequeñas preguntas, sino de aquellos problemas que integren todo el contenido desarrollado y debe responder a problemáticas lo más cercanas posible a la realidad circundante y a la práctica social. (7)

1.3. Evaluación del aprendizaje en la educación superior

La evaluación del aprendizaje tiene como propósito comprobar el grado de cumplimiento de los objetivos formulados en los planes y programas de estudio de la educación superior, mediante la valoración de los conocimientos y habilidades que los estudiantes van adquiriendo y desarrollando; así como, por la conducta que manifiestan en el proceso docente educativo.

“La evaluación del aprendizaje se estructura de forma frecuente, parcial, final y de culminación de los estudios, en correspondencia con el grado de sistematización de los objetivos a lograr por los estudiantes en cada momento del proceso. Estas formas de conjunto, caracterizan a la evaluación como un sistema. En correspondencia con su carácter continuo, cualitativo, integrador y basado fundamentalmente en el desempeño del estudiante, la tendencia que debe predominar en el sistema de evaluación es a que el peso fundamental de la misma descansa en las actividades evaluativas frecuentes y parciales, así como en evaluaciones finales de carácter integrador.” (8)

La evaluación frecuente tiene como propósito fundamental comprobar el grado de cumplimiento de los objetivos específicos en la ejecución del proceso docente educativo, mediante la valoración del trabajo de los estudiantes en todas las formas organizativas del proceso. Los tipos de evaluación frecuente a utilizar, por su gran versatilidad, se definen por el profesor. Los más utilizados son: las preguntas escritas y orales,

las discusiones grupales, entre otros. Por sus características, esta forma de evaluación constituye un elemento esencial de la evaluación del aprendizaje. (8)

La evaluación parcial tiene como propósito fundamental el logro de los objetivos de uno o varios temas. Los tipos fundamentales son la prueba parcial, el trabajo extraclase y el encuentro comprobatorio. El contenido de las evaluaciones parciales debe estar orientado a valorar, en diferentes momentos del proceso docente, las posibilidades de cada estudiante de aplicar los conocimientos adquiridos en la solución de problemas. (8)

La evaluación final tiene como propósito fundamental comprobar el grado de cumplimiento de los objetivos generales de una asignatura o disciplina. Sus tipos fundamentales son el examen final, la defensa del trabajo de curso y la evaluación final de la práctica laboral. El examen final es un tipo de evaluación final que comprueba los objetivos generales de las asignaturas o disciplinas. Evaluará, fundamentalmente, los conocimientos y las habilidades adquiridos por el estudiante, y su capacidad para integrarlos, sistematizarlos, aplicarlos y generalizarlos. (8)

1.4. Funciones de la evaluación

Las funciones de la evaluación son:

a) Función instructiva: Las distintas actividades de evaluación constituyen valiosas experiencias de aprendizaje para los alumnos, mediante las mismas, éstos infieren qué es lo más importante, de qué manera deben demostrar sus conocimientos y habilidades y sobre todo, consolidan el contenido de enseñanza apropiado en las distintas clases. La función instructiva tiene gran importancia pues contribuye al perfeccionamiento de los conocimientos. (9)

b) Función educativa: Esta función expresa la relación de la evaluación con las motivaciones de los estudiantes hacia el estudio. El conocimiento por parte de los alumnos de los resultados de la evaluación contribuye a que estos puedan trazarse una estrategia para erradicar las deficiencias, sirve de estímulo y a la vez posibilita una participación más consciente en el proceso de enseñanza. (9)

c) Función de diagnóstico: Permite el análisis de las causas que incidieron en las deficiencias detectadas en la evaluación. Estos resultados deben servir para que el profesor se formule múltiples interrogantes en

relación con las causas de los problemas que se hicieron notorios. Después de un proceso evaluativo, todo profesor debe tomar decisiones con respecto a los objetivos que no han sido alcanzados por los alumnos. (9)

d) Función de desarrollo: La evaluación debe contribuir al desarrollo intelectual, moral, político e ideológico de los alumnos. Esto exige que los aspectos incluidos en los exámenes tengan en cuenta el desarrollo independiente y creador de los alumnos y sus convicciones. (9)

e) Función de control: Esta función va más allá del trabajo del profesor en su clase, pues no se trata de recibir información y actuar conscientemente dentro de los límites de la enseñanza. De lo que se trata es de poner esa información en función de establecer estrategias más amplias por parte de los organismos estatales, para conocer la eficiencia del sistema de enseñanza y educación. Entran en juego no solo el análisis de una asignatura, sino también el Plan de Estudio y hasta el perfil del especialista, como en el caso de los centros de enseñanza superior. (9)

1.5. Cualidades de los instrumentos de evaluación

La elaboración de un instrumento de evaluación exige tener en cuenta tres cualidades fundamentales: validez, confiabilidad y facilidad de empleo.

La primera de estas cualidades, la validez, se define como la adecuada correspondencia entre los aspectos que se quieren evaluar y los que realmente se evalúan, esto significa que el contenido del examen debe ser representativo del universo de conocimientos y habilidades desarrollados en los alumnos. (9)

La confiabilidad consiste en la consistencia de los resultados de la aplicación del instrumento, pues este debe ser el mismo en caso de ser aplicado por otro profesor. (9)

La tercera cualidad, facilidad de empleo, se refiere a la necesidad de que los exámenes estén redactados en términos precisos y claros, cuya función es evitar subjetivismos e imprecisiones. (9)

La pregunta actúa como elemento básico del control en los exámenes, es importante tener en cuenta el tipo de pregunta y los requisitos al formularla. Las preguntas deben estar redactadas en lenguaje claro y al

alcance del estudiante, no deben ser extensas, ni muy amplias y no deben expresar más de una intención, de esta forma responden a la tercera cualidad de los instrumentos de evaluación.

Se clasifican atendiendo a los niveles de asimilación en:

Reproductivas: Comprueban comprensión consciente del objeto del conocimiento, son aquellas que exigen que el estudiante sea capaz de repetir el contenido que se le ha informado, ya sea este en forma declarativa o resolviendo problemas iguales o muy similares a los ya resueltos. Ejemplos: explicar un concepto, describir características, identificar, distinguir, reconocer, enunciar, definir. (10)

Aplicativas: Son preguntas para comprobar si los estudiantes son capaces de utilizar los conocimientos en la práctica. De tal forma cuando el estudiante resuelve problemas cuya situación le es desconocida y que exige que él conciba el modo de su solución, se está ante un nivel aplicativo. Ejemplos: comparar, clasificar, valorar importancia de procesos, relacionar. (10)

Creativas: Son preguntas para un nivel intelectual más alto. En este nivel creativo el estudiante tiene que hacer aportes cualitativamente novedosos para él, utilizando para ello, la lógica de la investigación científica. Ejemplos: formular hipótesis, elaborar un proyecto.(10)

1.6. Sistemas existentes para confeccionar instrumentos evaluativos en el ámbito internacional

Respondus

Respondus es una aplicación diseñada para Windows que permite la creación y edición de cuestionarios, para ser impresos en papel o publicados directamente en diferentes sistemas *elearning* (*Blackboard*, *eCollege*, *WebCT*, etc.). Los cuestionarios pueden crearse *offline* a través de un entorno de ventanas sencillo. En la pantalla de inicio, el programa muestra las diferentes acciones que se pueden llevar a cabo. Se pueden crear bases de datos de preguntas sobre el mismo programa, importar preguntas desde otros archivos (como *.txt*, *Microsoft Word*) siempre y cuando se presenten en el formato aceptado por Respondus, crear cuestionarios con preguntas previamente introducidas, o guardar un cuestionario con todos sus componentes (imágenes, ecuaciones, etc.) en un archivo comprimido. Respondus ofrece una gran versatilidad en lo que respecta al tipo de preguntas que puede crear. En función de la plataforma

seleccionada, se pueden crear un tipo de preguntas u otras, aunque, en general, la mayoría de ellas son coincidentes. (11)

Básicamente, existen ocho tipos de preguntas diferentes, entre las que se encuentran:

- Opción múltiple, con muchas respuestas posibles pero solo una respuesta correcta.
- Ensayo, para respuestas de desarrollo.
- Relacionar, para asociar conceptos o ideas.
- Rellenar espacios en blanco.
- Múltiples respuestas, similar al caso de opción múltiple, pero con más de una respuesta correcta.
- Ordenar.
- Calcular.

Hot Potatoes

Es una herramienta de creación de preguntas “*offline*” para educación desarrollada por Half Baked que permite crear preguntas del tipo selección múltiple, respuesta corta, crucigramas entre otras. *Hot Potatoes* no es un programa gratuito, su uso solo es gratuito para particulares y para instituciones educativas sin ánimo de lucro a condición de que los materiales que se elaboren con este programa se distribuyan gratuitamente a través de la *web* a toda persona interesada. Las herramientas que contiene el programa para crear las preguntas son:

JCLOZE: Genera un texto con huecos en blanco, donde se introduce las palabras que faltan.

JQUIZ: Genera una serie de preguntas y se introduce la respuesta en un cuadro de texto.

JCROSS: Genera crucigramas con espacios para introducir las respuestas.

JMIX: Genera ejercicios de ordenar frases.

JMATCH: Genera ejercicios de asociación.

The Masher: Permite compilar los ejercicios de *Hot Potatoes* en unidades didácticas.

Cualquier usuario sin grandes conocimientos de HTML, pero con algunos fundamentos básicos de informática puede manejarlo en corto tiempo, creando páginas dinámicas que pueden colocarse en la *web*. En este caso se necesita estar conectado a la red. Aunque no es un programa de última generación, su facilidad de uso, versatilidad y características dinámicas añadidas, han hecho de *Hot Potatoes* uno de los programas más populares en Internet en el ámbito educativo, con miles de usuarios en más de ciento sesenta países. (12)

Moodle

MOODLE es una herramienta que proporciona cursos basados en Internet y páginas *Web*. Fue diseñado por Martín Dougiamas de Perth. La palabra MOODLE, en inglés, es un acrónimo para Entorno de Aprendizaje Dinámico Orientado a Objeto y Modular. (13)

Una de las características principales de Moodle es su modularidad. Cada uno de los recursos que componen la plataforma se define como una pieza de *software* independiente. Los módulos son componentes auto-controlados que extienden las funcionalidades de una aplicación. Estos deben ser fáciles de instalar y mantener, asegurando aislar el impacto que pueda tener un fallo sobre el resto del programa. La lógica del funcionamiento de Moodle se encierra en tres grandes grupos de módulos: comunicación, recursos y actividades. (13)

Módulos de comunicación: Permiten que los alumnos puedan comunicarse con el profesor (hacer preguntas, plantear dudas) y entre ellos construir su propia comunidad de aprendizaje.

Módulos de recursos: Representan los contenidos y materiales del curso. Son todo tipo de textos, libros, apuntes, presentaciones de diapositivas, enlaces a páginas *web* externas, pensados para que los estudiantes los lean y estudien sobre ellos.

Módulos de actividades: Son la parte activa y colaborativa donde el estudiante tiene que hacer algo más que leer un texto. Debates y discusiones, resolución de problemas propuestos, redacción de trabajos, talleres, cuestionarios en línea.

Uno de los más amplios e importantes es el Módulo de Cuestionario que permite evaluar el desempeño de los estudiantes durante un curso, un semestre u otro período de tiempo determinado por los profesores

encargados del mismo. Se brinda la posibilidad de elaborar cuestionarios, compuestos por preguntas de diferentes tipos: opción múltiple, verdadero/falso, enlazar. Estos se conservan en la base de datos asociada a la plataforma, por lo que pueden ser reutilizados dentro del mismo curso o en otros. Los cuestionarios pueden permitir múltiples intentos, cada intento se califica automáticamente y el profesor puede decidir si muestra la calificación o las respuestas correctas a los alumnos, una vez concluido el cuestionario. (13)

1.7. Sistemas existentes para confeccionar instrumentos evaluativos en el ámbito nacional

GEXMEDI

Es un *software* que genera automáticamente cuestionarios para los Exámenes Estatales (EE) de la Carrera de Medicina. El sistema presenta un banco de preguntas, en el que los objetivos responden a los principales contenidos de las 4 grandes especialidades médicas (Pediatría, Cirugía, Medicina Interna, Ginecobstetricia) y también se incorpora la especialidad Medicina General Integral, como parte de la formación del recién graduado que, en su gran mayoría, va a ejercer la Medicina Familiar. Los exámenes generados son de utilidad para los profesores de pre y postgrado. El lenguaje de programación utilizado es *Microsoft Visual Basic 6.0* con el Gestor de Base de Datos *Microsoft Access 97*. (14)

Entre las características fundamentales del *software* se encuentran:

- Se dispone de la estructura de una base de datos con preguntas validadas, adaptable para cualquier asignatura.
- Una base de datos actualizada y ordenada con preguntas pertenecientes a las principales especialidades médicas.

Generador aleatorio de exámenes

Es una aplicación educativa (medio de enseñanza) que permite la generación y evaluación de exámenes con múltiples variantes en diferentes asignaturas en el Instituto Superior Pedagógico para Educación Técnica y Profesional (ISPETP). Permite generar de forma aleatoria o selectiva preguntas con diferentes niveles de asimilación del conocimiento; almacenar, imprimir, administrar, actualizar y evaluar exámenes. Cuenta con un repositorio de preguntas para exámenes en la asignatura que puedan ser utilizado en el proceso docente educativo, vinculadas al componente evaluación, que profesionaliza el proceso y

favorece el trabajo colaborativo entre profesores y especialistas. *Borland Delphi 7.0* fue la herramienta seleccionada para la programación de la aplicación. Tiene una base de datos creada en *Access* con la siguiente estructura:

- ✓ Tablas con información asociadas a las asignaturas propuestas.
- ✓ Tabla contraseña asociada a la clave de acceso a la aplicación.
- ✓ Tabla traza almacena información sobre evaluación del estudiante. (10)

1.8. Análisis de los sistemas estudiados para confeccionar instrumentos evaluativos

El estudio realizado sobre algunos de los sistemas que son utilizados para la elaboración de instrumentos de evaluación tanto a nivel nacional como internacional, evidenció que dichos *software* presentan poca posibilidad de ajustarlos a las necesidades del cliente.

En la Tabla 1 se comparan los distintos sistemas analizados en base a cinco parámetros: licencia de uso, tipo de aplicación, exportación a formatos para la impresión, preguntas tipo test y criterio de expertos:

Sistemas analizados	Licencia de uso		Tipo de aplicación		Exportan a formatos para la impresión	Preguntas tipo test	Criterio de expertos
	Libre	Propietaria	Web	Escritorio			
Respondus		x		x	x	x	
HotPotatoes		x		x		x	
Moodle	x		x			x	
GEXMEDI	x			x	x	x	
Generador aleatorio	x			x	x		

TABLA 1. COMPARACIÓN ENTRE LOS SISTEMAS ESTUDIADOS

Las herramientas Respondus y Hotpotatoes son de carácter propietario, lo que implica el pago de una licencia para su adquisición.

Respondus solo permite la elaboración de exámenes tipo *test* y no garantiza la seguridad de los exámenes porque no permite la gestión de usuarios y contraseñas.

Hot Potatoes genera páginas Web para propósitos educativos, pero de ninguna forma asegura que el alumno que supera determinado examen ha realizado el problema sin cometer ningún tipo de fraude, pues el código que implementa la funcionalidad está escrito en *JavaScript*, el cual tiene las siguientes características: se interpreta y se ejecuta en la máquina del usuario, esto significa que cualquiera podría comprobar cuál es el código e instrucciones que hacen funcionar la web y por tanto, podría averiguar cuáles son las respuestas correctas y así obtener la máxima calificación en los ejercicios, lo que evidencia que esta herramienta tiene grandes vulnerabilidades en la seguridad, el sistema solo permite la elaboración de cuestionarios tipo *test* y no los genera en formatos que faciliten su impresión.

En el caso de Moodle, la UCI cuenta con el Entorno Virtual de Aprendizaje (EVA), desarrollado sobre esta plataforma, dicho entorno puede ser accedido a través de la dirección <http://eva.uci.cu>. Cada curso en el EVA tiene cuestionarios para evaluar a los estudiantes, sin embargo estos no pueden generarse en formatos que faciliten su impresión y son elaborados con preguntas tipo *test*.

El análisis de los sistemas a nivel nacional, a pesar de no presentar limitante por pago de licencia, fueron desarrollados con tecnologías de poco uso en el desarrollo de *software* actual. Ambas carecen de seguridad en la base de datos, ya que el gestor utilizado fue *Microsoft Access 97* y a los productos de la *Suit Ofimática Office* se les han encontrado vulnerabilidades que permiten la obtención de las contraseñas de cifrado y la modificación del contenido de las mismas.

Ninguno de los sistemas estudiados permite crear un criterio de experto que categorice las preguntas por los niveles de complejidad.

A pesar de las deficiencias descritas de las herramientas analizadas, el estudio de estos sistemas arrojó características que permitieron definir valores agregados al desarrollo de la aplicación a desarrollar. Entre las que se pueden destacar:

- Clasificar las preguntas por niveles de dificultad.
- Gestionar los temas y objetivos a evaluar.

- Etiquetar las preguntas ya evaluadas, para que no se generen en otros temarios.
- Permitir el acceso al sistema a través de un usuario y una contraseña, con diferentes tipos de roles dentro del mismo.

1.9. Herramientas, técnicas y tecnologías a utilizar

1.9.1. Algoritmos criptográficos

Un algoritmo criptográfico, es un método que se usa en el proceso de cifrado y descifrado de información. Estos algoritmos, mezclan los datos en conjunto con una clave específica, llamados datos cifrados. Por otra parte, mediante el proceso inverso se reordena la información, así el algoritmo realiza el cálculo combinando los datos cifrados y su correspondiente clave, obteniendo de esta forma los datos descifrados.

Existen dos clases de algoritmos de encriptación basados en llaves o claves, Algoritmos Simétricos (o de llave secreta) y Algoritmos Asimétricos (o de llave pública). Los algoritmos simétricos utilizan la misma clave para encriptar y desencriptar o la llave de desencriptación es derivable de la llave de encriptación, mientras que los algoritmos asimétricos utilizan una llave distinta para la encriptación y para la desencriptación, y ninguna de las llaves puede ser derivada a partir de la otra. (15)

En este caso, los algoritmos simétricos fueron los escogidos por la seguridad, la rapidez y robustez que proveen. A continuación se detallan algunos de los algoritmos más utilizados.

Data Encryption Standard (DES)

DES (*Data Encryption Standard*) es un esquema de cifrado simétrico desarrollado en 1977 por el Departamento de Comercio y la Oficina Nacional de Estándares de Estados Unidos en colaboración con la empresa IBM.

DES utiliza una llave simétrica de 64 bits, de los cuales 56 son usados para el cifrado, mientras que los 8 restantes son de paridad, y se usan para la detección de errores en el proceso.

Los principales inconvenientes que presenta DES son:

- La llave es corta, tanto que no asegura una fortaleza adecuada. Con la potencia de cálculo actual de los computadores y con el trabajo en equipo por Internet se cree que se puede violar el algoritmo, como ya ha ocurrido, aunque en un plazo de tiempo que no resultó peligroso para la información cifrada.
- No permite longitud de llave variable, con lo que sus posibilidades de configuración son muy limitadas, además de permitirse con ello la creación de limitaciones legales.
- La seguridad del sistema se ve reducida considerablemente si se conoce un número suficiente textos elegidos, ya que existe un sistema matemático, llamado Criptoanálisis Diferencial, que puede en ese caso romper el sistema en 2^{47} iteraciones.

Entre sus ventajas se puede citar:

- Es el sistema más extendido del mundo y el más probado.
- Es muy rápido y fácil de implementar. (15)

IDEA

El algoritmo IDEA (*International Data Encryption Algorithm*) es más joven que DES, pues data de 1992. Trabaja con bloques de 64 bits de longitud y emplea una clave de 128 bits. Como en el caso de DES, se usa el mismo algoritmo tanto para cifrar como para descifrar. IDEA es un algoritmo bastante seguro, y hasta ahora se ha mostrado resistente a multitud de ataques, entre ellos el criptoanálisis diferencial. No presenta claves débiles, y su longitud de clave hace imposible en la práctica un ataque por la fuerza bruta. (15)

AES

Advanced Encryption Standard (AES), también conocido como Rijndael, es un esquema de cifrado por bloques adoptado como un estándar por el gobierno de los Estados Unidos.

Es un algoritmo simétrico de cifrado por bloques, donde las longitudes del bloque y de la llave son variables. Los bloques adoptados por el estándar son de 128 bits (no de 64 *bits* como el DES) y las llaves de 128, 192 o 256 *bits* (no 56 bits como el DES o 112 bits como el Triple DES). (15)

1.9.2. Metodología de desarrollo

Una metodología es un proceso que engloba procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de *software*. Se va indicando paso a paso lo que se debe hacer para lograr un producto informático. Además se debe especificar las personas que van a participar en el proceso así como el papel que van a jugar en él.

Existen dos tipos de metodologías: las Robustas y las Ágiles, entre los ejemplos del tipo de metodologías Robustas se encuentran: *Rational Unified Process* (RUP), *Microsoft Solutions Framework* (MSF) y Métrica 3.0, estas se recomiendan en proyectos complejos y de larga duración, obteniendo resultados satisfactorios cuando el equipo de trabajo tiene experiencia en su aplicación. La corrección de errores es muy costosa por ser metodologías poco flexibles a los cambios, debido a que el cliente y el equipo de desarrollo no mantienen una relación directa, no siendo el caso de la presente investigación.

Con la utilización de estas metodologías se genera una documentación amplia generando artefactos necesarios e innecesarios en algunos casos. Estas características hacen engorroso el desarrollo de proyectos pequeños, ralentizando su implementación y generando documentación que no será utilizada. Las metodologías robustas se guían por una fuerte planificación, centrando toda la atención en la documentación exhaustiva de los procesos llevados a cabo en el desarrollo, así como en las herramientas y notaciones que se utilizarán y en cumplir el plan del proyecto. (16)

Para el desarrollo de la herramienta quedó descartada la selección de una metodología pesada, dado que sus características no se adaptan a los requisitos de la investigación. Entre las metodologías Ágiles se encuentran: *Extreme Programming* (XP), *Scrum* y *Crystal*. Estas hacen menos énfasis en la documentación, puesto que el funcionamiento y el desarrollo del *software* son más importantes que la misma. La regla a seguir es no realizar documentos a menos que sean necesarios de forma inmediata, el cliente es muy importante y también son muy flexibles ante los cambios y no siguen estrictamente un plan. En cuanto a metas a seguir las metodologías ágiles presentan los siguientes principios:

- I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de *software* que le aporte un valor.
- II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.

- III. Entregar frecuentemente *software* que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- V. Construir el proyecto en torno a individuos motivados.
- VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo
- VII. El *software* que funciona es la medida principal de progreso.
- VIII. Los procesos ágiles promueven un desarrollo sostenible.
- IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- X. La simplicidad es esencial.
- XI. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- XII. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento. (17)

La propuesta de solución a desarrollar posee características que hacen que se ajuste a un proceso de desarrollo ágil, pues:

1. Se cuenta con un corto período de tiempo para el desarrollo.
2. Los requisitos pueden sufrir cambios constantemente a lo largo del proceso de desarrollo.
3. El cliente forma parte del equipo de desarrollo.
4. El equipo de desarrollo posee un tamaño pequeño.
5. La dimensión de la propuesta de solución es pequeña.
6. Se necesita realizar liberaciones frecuentes, para ser aprobadas por el cliente y recibir retroalimentación del mismo, lo cual requiere de un proceso de desarrollo preferiblemente ágil.

En la Tabla 2 se comparan distintas aproximaciones ágiles en base a tres parámetros: vista del sistema como algo cambiante, tener en cuenta la colaboración entre los miembros del equipo y características más específicas de la propia metodología como son simplicidad, excelencia técnica, resultados, adaptabilidad y prácticas de colaboración.

	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	5	4	3	3	4	5	5
Colaboración	5	5	4	4	4	5	5
Características Metodología (CM)							
-Resultados	5	5	4	4	4	5	5
-Simplicidad	4	4	3	5	3	5	5
-Adaptabilidad	5	5	3	3	4	4	3
-Excelencia técnica	3	3	4	4	4	3	4
-Prácticas de colaboración	5	5	4	3	3	4	5
Media CM	4.4	4.4	3.6	3.8	3.6	4.2	4.4
Media Total	4.8	4.5	3.6	3.6	3.9	4.7	4.8

TABLA 2 RANKING DE "AGILIDAD" (18)

Dentro de las metodologías ágiles analizadas XP destaca como una de las que mayor índice de agilidad posee. Resalta también por contar con la mayor cantidad de información disponible y es con diferencia la más popular.

Programación Extrema (XP, por sus siglas en inglés)

La programación extrema es una metodología de desarrollo ligero (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas. (17)

A continuación se presentan algunas características de XP que se adaptan a las necesidades de un proyecto, así como a las condiciones de trabajo:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas. Se aconseja escribir el código de la prueba antes de la codificación.

- Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera es revisado y discutido mientras se escribe.
- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores sean detectados.
- Simplicidad en el código: la programación extrema apuesta que es más sencillo hacer un código simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar funciones complicadas y quizás nunca utilizarla. (17)

El ciclo de vida ideal de XP consta de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto. (17)

Para una mejor estructuración del documento y un mejor entendimiento, se decidió agrupar las seis fases en tres, sin violar el ciclo de vida de la metodología, estas fases son:

- Fase I: Exploración
- Fase II: Planificación
- Fase III: Iteraciones

Se seleccionó XP como metodología de desarrollo porque propone existe una retroalimentación continua entre el cliente y el equipo de desarrollo, que constituye un factor muy importante porque conduce el desarrollo del trabajo y planifica el tiempo de entrega de cada iteración, el *software* será desarrollado por 2 programadores lo cual coincide con la característica de la metodología XP programación por pareja.

1.9.3. Lenguaje de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser ejecutados por máquinas computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. (19)

C#

C# es un lenguaje orientado a objetos, que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas. Su sintaxis es muy expresiva, sencilla y fácil de aprender, está basada en signos de llave, por lo que puede ser reconocida inmediatamente por cualquier persona familiarizada con C, C++ o Java, simplifica muchas de las complejidades de C++ y ofrece funciones tales como tipos de valores que aceptan datos *NULL*, enumeraciones, métodos anónimos y acceso directo a memoria, que no se encuentran en Java. (20)

En la plataforma .NET es posible programar en casi todos los lenguaje, C# es el lenguaje de propósito general diseñado por *Microsoft* para ser utilizado en ella. Por lo que se puede inferir en que si es el lenguaje nativo creado para esa plataforma, es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros. El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales a la propia fuente tales como ficheros de cabecera. El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile, lo que facilita la portabilidad del código. (20)

Las principales características de C# son:

Orientación a objetos: Como todo lenguaje de programación de propósito general actual, es un lenguaje orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código. (20)

Seguridad de tipos: incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura. (20)

Eficiente: todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad de procesamiento muy grande. (20)

1.9.4. Entornos Integrados de Desarrollo

Un entorno integrado de desarrollo, conocido también como IDE de sus siglas en inglés, es un programa informático compuesto por un conjunto de herramientas de programación, que puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE se deben seleccionar en dependencia de los lenguajes en que se piensa desarrollar, puesto que no todos los IDE soportan estos lenguajes candidatos. (21)

Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 es un entorno de desarrollo integrado para sistemas operativos *Windows*. Soporta varios lenguajes de programación tales como *Visual C++*, *Visual C#*, *Visual J#*, *ASP.NET* y *Visual Basic .NET*, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. (22)

Visual Studio 2010 es un conjunto completo de herramientas de gestión del ciclo de vida de una aplicación que garantizan resultados de calidad, desde el diseño hasta la implementación. Crea nuevas soluciones o mejora las aplicaciones existentes y permite realizar proyectos en un número creciente de plataformas y tecnologías, incluyendo la nube y la computación paralela. (23)

1.9.5. Gestor de Bases de Datos

Los sistemas de gestión de base de datos (SGBD) son un tipo de *software* muy específico, dedicados a servir de interfaz entre la base de datos, el usuario y las aplicaciones utilizadas. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. El propósito general es manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. (24)

Microsoft SQL Server 2008

El Servidor *Microsoft* para lenguaje estructurado de consultas (*Microsoft SQL Server*), por su nombre en inglés, es una plataforma de base de datos y análisis de datos que se utiliza en el procesamiento de transacciones en línea a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico. Ofrece la tecnología y las funciones con las que pueden contar las organizaciones. Con avances significativos en áreas clave de la administración de datos empresariales, la productividad de los desarrolladores y la inteligencia empresarial, las ventajas de este gestor son considerables: aprovechamiento de los activos de datos, aumento de la productividad y reducción de la complejidad de la tecnología de la información. Al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de *Microsoft*. (25)

Microsoft SQL Server 2008 usa los lenguajes de *.NET Framework*, además del lenguaje de programación *Transact-SQL* para crear objetos de base de datos. En proyectos de *Visual C#* se pueden crear procedimientos almacenados, desencadenadores, agregados, funciones y tipos definidos por el usuario. Todas estas funcionalidades más la propia integración con el IDE de desarrollo *Visual Studio 2010* lo convierte en el Gestor de Base de Datos idóneo para el sistema a desarrollar.

1.9.6. Lenguaje de modelado

En el proceso de desarrollo de un *software*, el modelado del mismo es de vital importancia y tiene una notable repercusión en su etapa de diseño e implementación, por proveer al ingeniero de un conjunto de notaciones, herramientas y prácticas, que le permiten visualizar el sistema a construir, logrando un nivel de abstracción que organice la lógica del mismo.

UML

UML (*Unified Modeling Language* por sus siglas en inglés), es el lenguaje de modelado más conocido y utilizado en la actualidad. Es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. Las propiedades que han hecho de UML un estándar son: la concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actual y futura. Reemplaza a decenas de notaciones empleadas con otros lenguajes y modela estructuras complejas. Las estructuras que soporta tienen sus fundamentos en las tecnologías orientadas a objetos, tales como clases, componentes y nodos. (26)

1.9.7. Herramientas para el diseño de la aplicación

Una CASE (de sus siglas en inglés *Computer Aided Software Engineering*, o Ingeniería de *Software* Asistida por Computadora) esencialmente, es una herramienta individual para ayudar al desarrollador o administrador de proyecto durante una o más fases del ciclo de desarrollo del *software* (o mantenimiento). (27)

Visual Paradigm 8.0

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite construir todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (28)

Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación.

Está diseñada para usuarios interesados en sistemas de *software* de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de *Java* y de UML. Incorpora el soporte para la labor en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. (29)

1.10. Conclusiones parciales

El estudio realizado en el capítulo sobre los sistemas para la confección de instrumentos evaluativos arrojó la necesidad de crear una aplicación que permita incluir en la elaboración de las preguntas a través de expertos la clasificación del nivel de complejidad, pues ninguno de los *software* analizados brindan esta posibilidad tan importante para aumentar la calidad de los instrumentos.

Se determinó a la metodología XP para guiar el proceso de desarrollo, pues se adapta perfectamente a las características del sistema, específicamente por los requisitos cambiantes, la programación en parejas y el cliente como un miembro más del equipo de desarrollo. Se decidió realizar una aplicación de escritorio seleccionando como lenguaje de programación C#, el IDE *Visual Studio* 2010 y el SGBD *Microsoft SQL Server* 2008, es válido aclarar que la selección estuvo determinada por las exigencias del cliente. El lenguaje de modelado es UML y como herramienta CASE para realizar el modelado del *software Visual Paradigm for UML* en su versión 8.0, aunque la metodología seleccionada fue XP y la misma no establece el uso del UML como un paso fundamental, sino una estrategia opcional, porque en su lugar esta metodología genera otros artefactos que facilitan mucho el trabajo del diseño del sistema. Para garantizar la seguridad de la base de datos del sistema el algoritmo de encriptación simétrico seleccionado es el AES debido a la estructura de su diseño.

CAPÍTULO 2. EXPLORACIÓN Y PLANIFICACIÓN

En la exploración los clientes plantean a grandes rasgos historias de usuarios que son de interés para la primera entrega del producto, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Durante la planificación el cliente establece la prioridad de cada historia de usuario y los programadores realizan una estimación del esfuerzo para implementarlas, se realiza el Plan de Iteraciones y el Plan de Entregas del producto.

2.1. Características de la propuesta de solución

La propuesta tiene como objetivo fundamental optimizar el proceso de confección de instrumentos evaluativos en la Facultad 2 de la UCI. Para ello se desarrollará una aplicación de escritorio donde se desempeñan 3 roles de usuarios: jefe de departamento, jefe de asignatura y profesor. La herramienta contará con 3 módulos específicos: Módulo de preguntas; Módulo de pruebas y Módulo de configuración.

El módulo de preguntas permitirá insertar, modificar, ver y eliminar preguntas de la base de datos. Las preguntas están estructuradas por las asignaturas, los temas, los objetivos y la complejidad a evaluar en éstas. Al módulo tienen acceso todos los usuarios del sistema.

Los niveles de complejidad de las preguntas se corresponden al nivel de asimilación del contenido, que se identifica por las habilidades que se pretende desarrollar en el estudiante. Un primer nivel de complejidad (bajo) con preguntas de carácter reproductivo que se asocia con las habilidades identificar, explicar o describir; un segundo nivel (medio) con preguntas de carácter aplicativo que se asocia con las habilidades comparar, clasificar, valorar; y un tercer nivel (alto) con preguntas creativas que son las preguntas para un nivel intelectual más alto. Las complejidades de las preguntas serán asignadas por los jefes de departamento y de asignaturas pues son los profesores de mayor experiencia en el claustro.

El módulo de pruebas permitirá crear, modificar, ver y eliminar instrumentos de evaluaciones. Los instrumentos evaluativos se pueden crear de forma manual o dinámicas. En el caso de la creación manual, los usuarios eligen las preguntas a aplicar, seleccionando todos los datos que la conforman y en el caso del dinámico, realiza un instrumento seleccionando la cantidad de preguntas que se desean, así como los objetivos, temas y la complejidad a evaluar por cada pregunta. Una vez creados los instrumentos

evaluativos el sistema permite a los usuarios guardarlos o generarlos a formato PDF, facilitando su impresión. El módulo puede ser accedido por todos los usuarios del sistema.

El módulo de configuración permitirá crear, modificar y eliminar los usuarios, asignaturas, temas de cada asignatura y objetivos, este módulo solo será accedido por el jefe del departamento y el jefe de asignatura. Los jefes de departamentos serán los encargados de la gestión de usuarios y las asignaturas, mientras que los jefes de asignatura gestionarán los datos referentes a temas y objetivos.

2.2. Usuarios relacionados con el sistema

Se denomina usuario a cualquier persona relacionada con el sistema, ya sea vinculada al desarrollo del mismo o que de una forma u otra interactúa con la aplicación.

Usuarios	Descripción
Jefe de departamento	El Jefe de departamento tiene acceso a todas las funcionalidades del sistema, excepto a la gestión de temas y objetivos.
Jefe de asignatura	El Jefe de asignatura tiene acceso a todas las funcionalidades del sistema, excepto a la gestión de usuarios y asignaturas.
Profesor	El profesor tiene acceso a las funcionalidades de los módulos preguntas y pruebas, excepto clasificar complejidad.

TABLA 3. USUARIOS RELACIONADOS CON EL SISTEMA

2.3. Exploración

En esta fase, el cliente describe con sus propias palabras las características del sistema que son más importantes para la primera entrega del producto. Después de que el cliente entregó la propuesta, el equipo de desarrollo analiza y define las dificultades técnicas que presentan cada una de estas características, así como su costo. Durante esta fase los programadores se familiarizan con las herramientas, tecnologías y prácticas que serán usadas en el desarrollo del proyecto y se exploran todas las posibilidades que puede tener la arquitectura del sistema. (30)

Se realizan tres operaciones fundamentales durante la exploración: escribir las historias de usuario, estimarlas y dividir historias en otras más pequeñas si se considera necesario. La duración de esta fase debe ser de pocas semanas a pocos meses, todo depende del tamaño del proyecto.

2.3.1. Historias de Usuarios

La metodología XP utiliza para representar los requisitos del sistema un artefacto denominado HU. Las mismas no son más que tarjetas de papel en las cuales el cliente describe las características del sistema. En cualquier momento del desarrollo pueden ser modificadas o reemplazadas por otras más generales o específicas, o incluso añadir nuevas. Cada una es entendible y está delimitada para que los desarrolladores puedan implementarlas en pocas semanas. (31)

No lograr determinar todas las HU desde un inicio no es un problema, puesto que al comienzo de cada iteración se registran los cambios en éstas y a partir de ahí se planifica la próxima iteración. Las HU se descomponen en tareas de ingeniería con una duración de uno a tres días, las cuales serán asignadas a los programadores para implementarlas durante una iteración. (30)

Según Kent Beck (32) cada HU recoge al menos los siguientes aspectos:

- **Número:** Posee el número asignado a la HU.
- **Nombre de HU:** Atributo que contiene el nombre de la HU.
- **Usuario:** El usuario del sistema que utiliza o protagoniza la HU.
- **Prioridad en el negocio:** Evidencia el nivel de prioridad de la HU en el negocio.
- **Riesgo de desarrollo:** Evidencia el nivel de riesgo en caso de no realizarse la HU.
- **Puntos estimados:** Este atributo no es más que una estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias. Por lo que, cuando el valor de dicho atributo es 0.3 equivale a 1 día y medio de trabajo, lo que se traduce aproximadamente en 15 horas.

- **Puntos reales:** Igual que el parámetro anterior, pero en este caso será el tiempo real en el que se realizó la HU.
- **Descripción:** Posee una breve descripción de lo que realizará la HU. (32)

Para las HU que describen las características del sistema para la elaboración de instrumentos de evaluaciones, se decidió agregarle un campo con el nombre **observación** que brindará información extra para que sean más comprensibles.

A continuación se muestran las HU del sistema.

Historia de Usuario	
Número: 1	Nombre: <i>Autenticar Usuario</i>
Usuario: <i>Profesor, Jefe de asignatura y Jefe de departamento</i>	
Prioridad en el negocio: <i>Alta</i>	Puntos estimados: 1
Riesgo de Desarrollo: <i>Alto</i>	Puntos reales: 1
Descripción: <i>Permitirá a los usuarios acceder al sistema a través de un usuario y una clave de acceso personal.</i>	
Observación:	

TABLA 4. HU AUTENTICAR USUARIO

Historia de Usuario	
Número: 2	Nombre: <i>Gestionar Preguntas</i>
Usuario: <i>Profesor, Jefe de asignatura y Jefe de departamento</i>	
Prioridad en el negocio: <i>Alta</i>	Puntos estimados: 1
Riesgo de Desarrollo: <i>Alto</i>	Puntos reales: 1
Descripción: <i>Los usuarios del sistema pueden gestionar las preguntas en el sistema. En el caso de las modificaciones y las preguntas a eliminar, son operaciones que los profesores realizarán sobre preguntas insertadas por ellos. El jefe de la asignatura puede eliminar o modificar cualquier pregunta en la base de datos.</i>	
Observación: <i>Para gestionar las preguntas el usuario debe seleccionar la asignatura, posteriormente selecciona el tema y el objetivo a evaluar. Una vez realizado estos pasos el usuario puede adicionar, eliminar, modificar o ver las preguntas.</i>	

TABLA 5. HU GESTIONAR PREGUNTA

Historia de Usuario	
Número: 3	Nombre: Gestionar pruebas
Usuario: Profesor, Jefe de asignatura y Jefe de departamento	
Prioridad en el negocio: Alta	Puntos estimados: 1
Riesgo de Desarrollo: Alto	Puntos reales: 1
Descripción: Los usuarios del sistema pueden gestionar instrumentos evaluativos en el sistema. El jefe de la asignatura y el Jefe de departamento tienen acceso pleno al módulo. Los profesores podrán crear y ver instrumentos evaluativos, así como editar y eliminar los instrumentos evaluativos realizados por ellos.	
Observación: La gestión de los instrumentos evaluativos permitirá al usuario crear, eliminar, modificar o ver los instrumentos evaluativos. Estos se pueden crear de forma dinámica o manualmente seleccionando las preguntas, con sus niveles de dificultad, objetivos y temas. Mientras la edición, la eliminación y el mostrar permitirán al usuario realizar estas acciones sobre el examen deseado.	

TABLA 6. HU GESTIONAR EXÁMENES

Historia de Usuario	
Número: 4	Nombre: Exportar exámenes a PDF
Usuario: Profesor, Jefe de asignatura y Jefe de departamento	
Prioridad en el negocio: Alta	Puntos estimados: 1
Riesgo de Desarrollo: Alto	Puntos reales: 1
Descripción: Una vez creado los ejercicios los usuarios pueden exportarlo a PDF para facilitar su impresión.	
Observación:	

TABLA 7. HU EXPORTAR EXÁMENES A PDF

Historia de Usuario	
Número: 5	Nombre: Asignar asignatura
Usuario: Jefe de departamento	
Prioridad en el negocio: Alta	Puntos estimados: 0.5
Riesgo de Desarrollo: Alto	Puntos reales: 0.5
Descripción: Se les asigna a los usuarios determinadas asignaturas.	
Observación: Una vez creado los usuarios por el Jefe de departamento, se le asigna a cada uno la asignatura o asignaturas que va a impartir.	

TABLA 8. HU ASIGNAR ASIGNATURA

Historia de Usuario	
Número: 6	Nombre: <i>Asignar complejidad</i>
Usuario: <i>Jefe de asignatura y Jefe de departamento</i>	
Prioridad en el negocio: <i>Alto</i>	Puntos estimados: <i>0.5</i>
Riesgo de Desarrollo: <i>Alto</i>	Puntos reales: <i>0.5</i>
Descripción: <i>Se le asigna a cada pregunta la complejidad correspondiente.</i>	
Observación: <i>Cuando se insertan las preguntas en el sistema, el Jefe de asignatura o el Jefe de departamento le asignan la complejidad a éstas.</i>	

TABLA 9. HU ASIGNAR COMPLEJIDAD

Historia de Usuario	
Número: 7	Nombre: <i>Gestionar usuario</i>
Usuario: <i>Jefe de departamento</i>	
Prioridad en el negocio: <i>Alta</i>	Puntos estimados: <i>1</i>
Riesgo de Desarrollo: <i>Alto</i>	Puntos reales: <i>1</i>
Descripción: <i>El Jefe de departamento puede crear, modificar y eliminar usuarios en el sistema.</i>	
Observación:	

TABLA 10. HU GESTIONAR USUARIO

Historia de Usuario	
Número: 8	Nombre: <i>Gestionar asignatura</i>
Usuario: <i>Jefe de departamento</i>	
Prioridad en el negocio: <i>Alta</i>	Puntos estimados: <i>1</i>
Riesgo de Desarrollo: <i>Alto</i>	Puntos reales: <i>1</i>
Descripción: <i>El Jefe de departamento puede crear, modificar y eliminar asignaturas en el sistema.</i>	
Observación:	

TABLA 11. HU GESTIONAR ASIGNATURA

Historia de Usuario	
Número: 9	Nombre: <i>Gestionar tema</i>
Usuario: <i>Jefe de asignatura y Jefe de departamento</i>	
Prioridad en el negocio: <i>Alta</i>	Puntos estimados: 1
Riesgo de Desarrollo: <i>Alto</i>	Puntos reales: 1
Descripción: <i>El Jefe de asignatura y el Jefe de departamento pueden crear, modificar y eliminar temas en el sistema.</i>	
Observación:	

TABLA 12. HU GESTIONAR TEMA

Historia de Usuario	
Número: 10	Nombre: <i>Gestionar objetivos</i>
Usuario: <i>Jefe de asignatura y Jefe de departamento</i>	
Prioridad en el negocio: <i>Alta</i>	Puntos estimados: 1
Riesgo de Desarrollo: <i>Alto</i>	Puntos reales: 1
Descripción: <i>El Jefe de asignatura y el Jefe de departamento pueden crear, modificar y eliminar temas en el sistema.</i>	
Observación:	

TABLA 13. HU GESTIONAR OBJETIVOS

Historia de Usuario	
Número: 11	Nombre: <i>Seguridad</i>
Descripción: <i>La autenticación en el sistema está definida para cada uno de los usuarios mediante su usuario y contraseña.</i>	
Observación: <i>La aplicación sólo podrá ser accedida por medio de un usuario y una contraseña. El usuario registrado podrá realizar las operaciones de acuerdo a su rol definido en el sistema.</i>	

TABLA 14. SEGURIDAD

Historia de Usuario	
Número: 12	Nombre: Usabilidad
Descripción: El sistema será usado por personas con pocos, medios o avanzados conocimientos de informática.	
Observación: El sistema muestra una interfaz sencilla y amigable. Brinda los botones con un tamaño adecuado y con nombres claros que permiten a los usuarios realizar las operaciones que deseen de forma sencilla.	

TABLA 15. USABILIDAD

Historia de Usuario	
Número: 13	Nombre: Condiciones tecnológicas
Descripción: El acceso al sistema se puede realizar desde ordenadores con bajas prestaciones.	
Observación: Las máquinas donde se van a utilizar la aplicación deben tener como mínimo 256 Mb de memoria RAM y como sistema operativo Windows XP o superior.	

TABLA 16. CONDICIONES TECNOLÓGICAS

2.4. Planificación

XP plantea la planificación como un constante diálogo entre el cliente y los desarrolladores, donde los primeros definen el alcance del proyecto y la prioridad con qué debe ser hecho en primer lugar. Toda técnica de planificación de *software* debe tratar de crear visibilidad, de tal manera que todo individuo involucrado en el proyecto pueda realmente ver cuán largo es el proyecto. (33)

En esta fase los desarrolladores y el cliente establecen los tiempos de implementación de cada HU, la prioridad con que serán desarrolladas y las historias que serán implementadas en cada versión del proyecto. Se realizará un plan de entregas y en cada iteración el cliente recibirá una nueva versión.

2.4.1. Estimación de esfuerzos por HU

Las HU deben tener el detalle mínimo para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogan directamente con el cliente para obtener todos los detalles necesarios. (33)

Inicialmente, el equipo de desarrolladores estima el esfuerzo necesario para implementar las HU y los clientes aprueban los objetivos y tiempos de entrega, la estimación temporal se basa en un cálculo

estimado por parte de los desarrolladores de cada una de las HU. Las estimaciones realizadas en esta fase son primarias y podrían variar cuando se analicen más en detalle en cada iteración. (33)

A continuación se muestra la tabla Estimación de esfuerzos:

No.	Historia de Usuario	Estimación (por semanas)
1	<i>Autenticar Usuario</i>	<i>1 semana</i>
2	<i>Gestionar Preguntas</i>	<i>1 semana</i>
3	<i>Gestionar Pruebas</i>	<i>1 semana</i>
4	<i>Exportar exámenes a PDF</i>	<i>1 semana</i>
5	<i>Asignar Asignatura</i>	<i>0.5 semana</i>
6	<i>Asignar complejidad</i>	<i>0.5 semana</i>
7	<i>Gestionar usuario</i>	<i>1 semana</i>
8	<i>Gestionar asignatura</i>	<i>1 semana</i>
9	<i>Gestionar tema</i>	<i>1 semana</i>
10	<i>Gestionar objetivos</i>	<i>1 semana</i>

TABLA 17. ESTIMACIÓN DE ESFUERZOS POR HU

2.5. Plan de iteraciones

Todo proyecto que emplea metodología XP debe dividirse en iteraciones. En el plan de iteraciones se especifican detalladamente el orden de desarrollo de las HU dentro de cada iteración conjuntamente con la duración de las mismas.

Iteración 1: En esta iteración se implementan las HU que tienen prioridad alta en el negocio, las principales son las HU 1, 2 y 3, las cuales son de vital importancia ya que conforman la base de la estructura del sistema. Con esta iteración se obtiene la primera versión de la aplicación la cual se utilizará para mostrar al cliente y permitirá al grupo de trabajo tener una retroalimentación.

Iteración 2: En esta iteración se realiza la implementación de las HU 4, 5 y 6. Con esta iteración se corrigen errores o inconformidades del cliente con las HU implementadas en la iteración anterior. De esta

forma se obtiene la segunda versión del sistema. Esta segunda iteración es también mostrada al cliente con el objetivo de evaluar las aceptaciones de su parte con el *software*.

Iteración 3: En esta iteración se realiza la implementación de la HU 7, 8, 9 y 10 corrigiéndose los errores de las iteraciones anteriores.

2.5.1. Plan de duración de iteraciones

El plan de duración de las iteraciones se realiza luego de tener el estimado en semanas que demora implementar cada historia de usuario. Su objetivo es especificar detalladamente el orden de desarrollo de las HU dentro de cada iteración y la duración de las mismas. (33)

Iteraciones	Orden de las HU a implementar	Duración (semanas)
Iteración 1	HU1. Autenticar Usuario HU2. Gestionar Preguntas HU3. Gestionar Pruebas	3 semanas
Iteración 2	HU4. Exportar exámenes a PDF HU5. Asignar Asignatura HU6. Asignar Complejidad	2 semanas
Iteración 3	HU7. Gestionar usuarios HU8. Gestionar asignatura HU9. Gestionar tema HU10. Gestionar objetivos	4 semanas

TABLA 18. PLAN DE DURACIÓN DE ITERACIONES

2.5.2. Plan de entregas

En el plan de entregas se establecen que HU son agrupadas para conformar una entrega y el orden de las mismas.

Historia de Usuario	Primera Iteración (1ra semana de marzo)	Segunda Iteración (1ra semana de abril)	Tercera Iteración (1ra semana de mayo)
Autenticar Usuario	V 1.0	Finalizado	Finalizado
Gestionar Preguntas	V 1.0	Finalizado	Finalizado
Gestionar pruebas	V 1.0	Finalizado	Finalizado
Exportar exámenes a PDF	-	V1.0	Finalizado
Asignar Asignatura	-	V1.0	Finalizado
Asignar Complejidad	-	V1.0	Finalizado
Gestionar Usuario	-	-	Finalizado
Gestionar Asignatura	-	-	Finalizado
Gestionar Tema	-	-	Finalizado
Gestionar Objetivos	-	-	Finalizado

TABLA 19. PLAN DE ENTREGAS

2.6. Conclusiones parciales

Durante el desarrollo de este capítulo se determinaron 3 tipos de usuarios que interactúan con el sistema: Jefe de departamento, Jefe de asignatura y Profesor. Se definieron 10 Historias de Usuarios para dar cumplimiento a la propuesta de solución, que serán implementadas en tres iteraciones con una duración de 9 semanas y el plan de entrega para liberar las versiones del sistema.

CAPÍTULO 3. ITERACIONES

El presente capítulo está dedicado a la fase implementación. Se presentan las tarjetas Clase-Responsabilidades-Colaboradores (CRC) y el modelo de datos a través de un Diagrama Entidad-Relación (DER). Se presenta una descripción de la arquitectura de la propuesta de solución, en este caso Modelo-Vista-Controlador y se realiza una descripción de los patrones de diseño y estándares de codificación utilizados para implementar la solución. También se muestran las tareas de ingeniería necesarias para llevar a cabo el proceso de desarrollo.

3.1. Diseño del sistema

Tarjetas CRC

Las tarjetas CRC son en la práctica pequeñas tarjetas de cartón que se elaboran para ser mostradas al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas, lo que ayuda al equipo durante el diseño e implementación del sistema. Las tarjetas CRC trabajan con la técnica de modelado basada en objetos, representando cada una a un objeto, identificando las clases y sus responsabilidades. Las tarjetas están compuestas por el nombre de la clase colocado como título, en la parte izquierda se colocan las responsabilidades (funcionalidades) y en la parte derecha las clases que se implican en cada funcionalidad. (35)

Tarjeta CRC	
Clase: Nombre de la clase que se está modelando.	
Súper Clase: Nombre de la clase padre en la herencia.	
Sub Clase (s): Nombre de la(s) clase(s) hija en la herencia.	
Responsabilidad: Es una descripción de alto nivel del propósito de la clase.	Colaboraciones: Indica con cuáles otras clases se requiere relación para cumplir la responsabilidad.

TABLA 20. MODELO DE LAS TARJETAS CRC

A continuación se muestran las tarjetas CRC de las clases controladoras correspondientes a la solución propuesta:

Tarjeta CRC	
Clase: Usuario	
Responsabilidad: Gestionar los diferentes usuarios de la aplicación, así como los datos que contienen.	Colaboraciones: Autenticar

TABLA 21. TARJETA CRC: USUARIO

Tarjeta CRC	
Clase: Autenticar	
Responsabilidad: Autenticar los usuarios que interactúan con el sistema.	Colaboraciones: Usuario

TABLA 22. TARJETA CRC: AUTENTICAR

Tarjeta CRC	
Clase: AES	
Responsabilidad: Es la encargada de encriptar y desencriptar los datos con los que se interactúan en la base de datos.	Colaboraciones: Conexión

TABLA 23. TARJETA CRC: AES

Tarjeta CRC	
Clase: PDF	
Responsabilidad: Es la encargada de exportar los exámenes en formato PDF.	Colaboraciones:

TABLA 24. TARJETA CRC: PDF

Tarjeta CRC	
Clase: Conexión	
Responsabilidad: Es la encargada del acceso a datos del sistema.	Colaboraciones: Autenticar, AES

TABLA 25. TARJETA CRC: CONEXIÓN

3.2. Diseño de la Base de Datos

En cualquier aplicación en la cual se gestione información, la base de datos desempeña un papel fundamental. El diseño de la misma es algo esencial para el desarrollo de sistemas de este tipo, ya que esta permite el almacenamiento de la información de forma coherente y organizada, para evitar pérdidas e

inconsistencias, de esta manera la recuperación de la información es de forma rápida y flexible. Por tal motivo enfocarse en su desarrollo es una tarea vital. A continuación se muestra el diagrama entidad - relación de la base de datos que se utilizó, el mismo fue generado de forma automática por la herramienta CASE *Visual Paradigm for UML*.

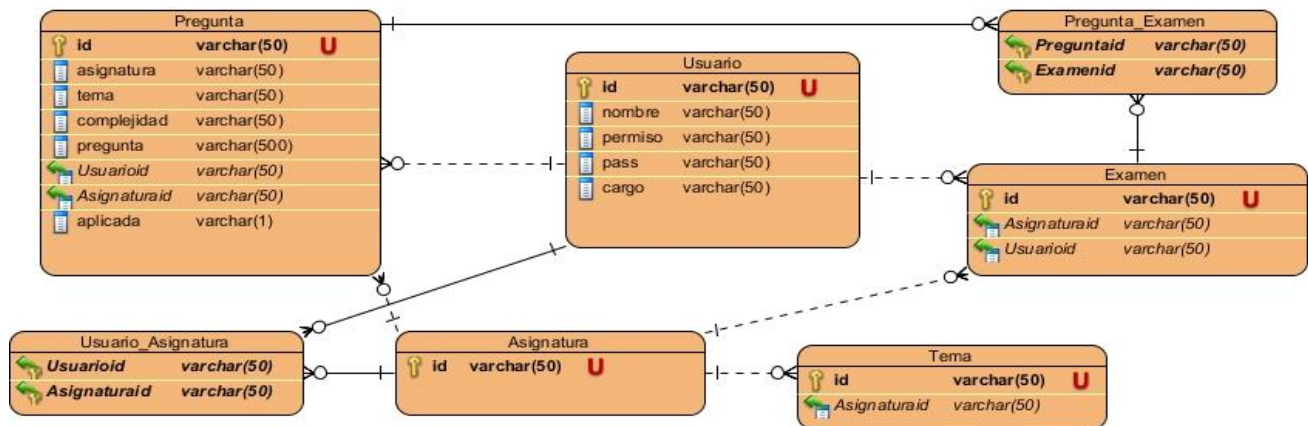


Figura 1. Diagrama Entidad Relación de la Base de Datos

3.3. Patrones de Arquitectura

Un patrón es un modelo a seguir, surge de la experiencia de seres humanos al tratar de lograr ciertos objetivos, captura la experiencia existente y probada para promover buenas prácticas. Los patrones arquitectónicos son los que definen la estructura de un sistema de *software*, los cuales a su vez se componen de subsistemas con sus responsabilidades. También tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño. (36)

Patrón Modelo Vista Controlador

El patrón Modelo Vista Controlador (MVC) es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. (37)

Modelo: Es el objeto que representa los datos del programa. Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y comportamiento de entrada.

Vista: Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

Controlador: Recibe las entradas, usualmente como eventos. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

3.4. Patrones de Diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. El mismo identifica: clases, instancias, roles, colaboraciones y la distribución de responsabilidades. (38)

En resumen los patrones de diseño se caracterizan por:

- Ser soluciones concretas.
- Ser soluciones técnicas.
- Se utilizan en situaciones frecuentes.
- Favorecen la reutilización de código.
- Su uso no se refleja en el código.

Patrones GRASP

Los patrones GRASP (Patrones Generales de *Software* para Asignar Responsabilidades) describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, si se desea diseñar un *software* orientado a objetos con calidad. (39)

En el diseño de la aplicación propuesta se utilizaron los patrones GRASP experto, creador, controlador y bajo acoplamiento.

Experto: Es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo.

Creador: Ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador.

Controlador: Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

Bajo acoplamiento: Mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Este patrón se tuvo presente debido a la importancia que se le atribuye a realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y que a su vez permitan la reutilización. (39)

3.5. Estilos y estándares de codificación

Durante el proceso de implementación de un *software* es considerado una buena práctica realizar la codificación del mismo siguiendo estándares que lo guíen. La metodología XP propone el uso de estándares de codificación, reforzando esta práctica con otras como la programación en parejas, la propiedad colectiva del código y la integración continua. De modo que el código no sea conocido por una sola persona del equipo de desarrollo, esto permite que otros miembros del equipo puedan realizar cambios en el código, y así poder integrar de forma continua los nuevos cambios, aun cuando por determinadas razones falle el factor humano. A continuación se muestran los estándares de codificación más relevantes utilizados en la implementación de la propuesta de solución.

1. Los comentarios multilíneas se escriben comenzando con los caracteres `"/**` y terminando con `*/`, los comentarios de una sola línea comienzan con los caracteres `//`.
2. Los nombres de las clases del modelo utilizan el estándar *UpperCamelCase* y las clases controladoras y los métodos utilizan en sus nombres el estándar *CamelCase*.
3. Todos los atributos de las clases del modelo están encapsulados. Son privados y el acceso a ellos se hace por medio de los métodos `getAtributo` y/o `setAtributo`.
4. Las variables son explícitas, aunque se usan abreviaturas que no violan este principio.

El uso de estándares facilitó en gran medida una mejor comprensión del código por los desarrolladores, durante la implementación. El código muestra una mayor limpieza, claridad y organización, elementos estos que aportan un valor agregado a la propuesta de solución.

3.6. Fase de Iteraciones

La parte más importante en la programación extrema es la implementación. La metodología XP sugiere un modelo de trabajo usando repositorios de código donde las parejas de programadores publican cada pocas horas lo implementado y corregido junto a los *test* que deben pasar. Propone un modelo de desarrollo colectivo en el que todos los programadores están implicados en todas las tareas. El permitirle a los programadores modificar códigos que no son suyos no supone ningún riesgo. La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar. (33)

La programación extrema propone una forma iterativa para la implementación de un *software* junto a las prácticas. Al terminar cada iteración se obtiene una versión del producto funcional que debe ser probado y mostrado al cliente.

Iteraciones

Para el desarrollo de la aplicación se definieron tres iteraciones donde se describe la funcionalidad a implementar y el responsable de cumplir con esta tarea, así como el tiempo que le va a dedicar a la misma. A continuación se detallan las tareas de desarrollo a realizar en cada una de las iteraciones previstas.

Primera Iteración

Esta primera iteración tiene como objetivo dar cumplimiento a las HU 1, 2 y 3 que representan mayor prioridad de desarrollo para el sistema. Al terminar esta iteración deben haber quedado implementadas las funcionalidades propuestas.

Tarea de Ingeniería	
Número de la tarea: 1	Número de la HU: 1
Nombre de la tarea: <i>Autenticar Usuario</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 1
Fecha de inicio: 10 de febrero de 2014	Fecha de fin: 15 febrero de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 26. TAREA DE INGENIERÍA #1

Tarea de Ingeniería	
Número de la tarea: 2	Número de la HU: 2
Nombre de la tarea: <i>Crear pregunta</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.2
Fecha de inicio: 17 de febrero de 2014	Fecha de fin: 18 de febrero de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 27. TAREA DE INGENIERÍA #2

Tarea de Ingeniería	
Número de la tarea: 3	Número de la HU: 2
Nombre de la tarea: <i>Editar pregunta</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.2
Fecha de inicio: 19 de febrero de 2014	Fecha de fin: 20 de febrero de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 28. TAREA DE INGENIERÍA #3

Tarea de Ingeniería	
Número de la tarea: 4	Número de la HU: 2
Nombre de la tarea: <i>Eliminar pregunta</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 20 de febrero de 2014	Fecha de fin: 21 de febrero de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 29. TAREA DE INGENIERÍA #4

Tarea de Ingeniería	
Número de la tarea: 5	Número de la HU: 2
Nombre de la tarea: <i>Ver pregunta</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 21 de febrero de 2014	Fecha de fin: 22 de febrero de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 30. TAREA DE INGENIERÍA #5

Tarea de Ingeniería	
Número de la tarea: 6	Número de la HU: 3
Nombre de la tarea: <i>Crear exámenes manualmente</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 23 de febrero de 2014	Fecha de fin: 25 de febrero de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 31. TAREA DE INGENIERÍA #6

Tarea de Ingeniería	
Número de la tarea: 7	Número de la HU: 3
Nombre de la tarea: <i>Crear exámenes aleatorios</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 25 de febrero de 2014	Fecha de fin: 27 de febrero de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 32. TAREA DE INGENIERÍA #7

Tarea de Ingeniería	
Número de la tarea: 8	Número de la HU: 3
Nombre de la tarea: <i>Modificar exámenes</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.1
Fecha de inicio: 27 de febrero de 2014	Fecha de fin: 28 de febrero de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 33. TAREA DE INGENIERÍA #8

Tarea de Ingeniería	
Número de la tarea: 9	Número de la HU: 3
Nombre de la tarea: <i>Eliminar exámenes</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.1
Fecha de inicio: 28 de febrero de 2014	Fecha de fin: 1 marzo de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 34. TAREA DE INGENIERÍA #9

Tarea de Ingeniería	
Número de la tarea: 10	Número de la HU: 3
Nombre de la tarea: <i>Ver exámenes</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.1
Fecha de inicio: 1 marzo de 2014	Fecha de fin: 2 marzo de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 35. TAREA DE INGENIERÍA #10

Tarea de Ingeniería	
Número de la tarea: 11	Número de la HU: 3
Nombre de la tarea: <i>Guardar exámenes</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.1
Fecha de inicio: 2 marzo de 2014	Fecha de fin: 3 marzo de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 36. TAREA DE INGENIERÍA #11

Segunda Iteración

Durante esta iteración se trataron las tareas correspondientes a las historias de usuarios 4, 5 y 6, estas serán mostradas al cliente con el objetivo de realizar cambios necesarios en base a la opinión del mismo.

Tarea de Ingeniería	
Número de la tarea: 12	Número de la HU: 4
Nombre de la tarea: <i>Exportar exámenes a PDF</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 1
Fecha de inicio: 11 de marzo de 2014	Fecha de fin: 16 de marzo de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 37. TAREA DE INGENIERÍA #12

Tarea de Ingeniería	
Número de la tarea: 13	Número de la HU: 5
Nombre de la tarea: Asignar asignatura	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.5
Fecha de inicio: 18 de marzo de 2014	Fecha de fin: 20 de marzo de 2014
Programador responsable: Félix Ramón Quintero Mora, Reinaldo Enrique Merino	

TABLA 38. TAREA DE INGENIERÍA #13

Tarea de Ingeniería	
Número de la tarea: 14	Número de la HU: 6
Nombre de la tarea: Asignar complejidad	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.5
Fecha de inicio: 20 de marzo de 2014	Fecha de fin: 23 de marzo de 2014
Programador responsable: Félix Ramón Quintero Mora, Reinaldo Enrique Merino	

TABLA 39. TAREA DE INGENIERÍA #14

Tercera Iteración

Durante esta iteración se trataron las tareas correspondientes a las historias de usuarios restantes. Al culminar esta, se consta de un producto listo para su puesta en funcionamiento.

Tarea de Ingeniería	
Número de la tarea: 15	Número de la HU: 7
Nombre de la tarea: <i>Agregar usuario</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 24 de marzo de 2014	Fecha de fin: 26 de marzo de 2014
Programador responsable: Félix Ramón Quintero Mora, Reinaldo Enrique Merino	

TABLA 40. TAREA DE INGENIERÍA #15

Tarea de Ingeniería	
Número de la tarea: 16	Número de la HU: 7
Nombre de la tarea: <i>Editar usuario</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 26 de marzo de 2014	Fecha de fin: 28 de marzo de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 41. TAREA DE INGENIERÍA #16

Tarea de Ingeniería	
Número de la tarea: 17	Número de la HU: 7
Nombre de la tarea: <i>Eliminar usuario</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 28 de marzo de 2014	Fecha de fin: 30 de marzo de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 42. TAREA DE INGENIERÍA #17

Tarea de Ingeniería	
Número de la tarea: 18	Número de la HU: 8
Nombre de la tarea: <i>Adicionar asignatura</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 7 de abril de 2014	Fecha de fin: 9 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 43. TAREA DE INGENIERÍA #18

Tarea de Ingeniería	
Número de la tarea: 19	Número de la HU: 8
Nombre de la tarea: <i>Editar asignatura</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 9 de abril de 2014	Fecha de fin: 11 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 44. TAREA DE INGENIERÍA #19

Tarea de Ingeniería	
Número de la tarea: 20	Número de la HU: 8
Nombre de la tarea: <i>Eliminar asignatura</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 11 de abril de 2014	Fecha de fin: 13 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 45. TAREA DE INGENIERÍA #20

Tarea de Ingeniería	
Número de la tarea: 21	Número de la HU: 9
Nombre de la tarea: <i>Adicionar tema</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 15 de abril de 2014	Fecha de fin: 17 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 46. TAREA DE INGENIERÍA #21

Tarea de Ingeniería	
Número de la tarea: 22	Número de la HU: 9
Nombre de la tarea: <i>Editar tema</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 17 de abril de 2014	Fecha de fin: 19 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 47. TAREA DE INGENIERÍA #22

Tarea de Ingeniería	
Número de la tarea: 23	Número de la HU: 9
Nombre de la tarea: <i>Eliminar tema</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 19 de abril de 2014	Fecha de fin: 21 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 48. TAREA DE INGENIERÍA #23

Tarea de Ingeniería	
Número de la tarea: 24	Número de la HU: 10
Nombre de la tarea: <i>Adicionar objetivo</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 21 de abril de 2014	Fecha de fin: 23 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 49. TAREA DE INGENIERÍA #24

Tarea de Ingeniería	
Número de la tarea: 25	Número de la HU: 10
Nombre de la tarea: <i>Editar objetivo</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 23 de abril de 2014	Fecha de fin: 25 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 50. TAREA DE INGENIERÍA #25

Tarea de Ingeniería	
Número de la tarea: 26	Número de la HU: 10
Nombre de la tarea: <i>Eliminar objetivo</i>	
Tipo de tarea: <i>Desarrollo</i>	Puntos estimados: 0.3
Fecha de inicio: 25 de abril de 2014	Fecha de fin: 27 de abril de 2014
Programador responsable: <i>Félix Ramón Quintero Mora, Reinaldo Enrique Merino</i>	

TABLA 51. TAREA DE INGENIERÍA #26

3.7. Pruebas

La metodología XP enfatiza en la realización de un sin número de pruebas a lo largo de todo el desarrollo del *software*, con el fin de lograr un producto con calidad, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su corrección. En este proceso no solo participa el equipo de desarrollo, también es importante la colaboración del cliente, sobre todo en las pruebas de aceptación. (33)

En XP las pruebas se dividen en dos grupos: pruebas unitarias encargadas de verificar el código y pruebas de aceptación que están orientadas a probar las funcionalidades del sistema. Un equipo de

desarrollo que siga el proceso de XP debe primero probar su código de forma unitaria, integrar este a la aplicación y después realizar las pruebas de aceptación.

3.7.1. Desarrollo dirigido por pruebas (TDD)

El desarrollo dirigido por pruebas es una característica de la metodología XP para comprobar el funcionamiento de los códigos que se van implementando. Este estilo sigue el principio “*test-first*”, significa que las pruebas se crean antes de comenzar la implementación. La idea es que, al pensar en cómo se probará la funcionalidad, se está pensando en la propia funcionalidad desde el punto de vista de su interfaz (qué métodos tendrá y con qué parámetros), ayudando a desarrollar así un mejor diseño. De ésta manera, además, se asegura de que no exista ninguna funcionalidad que no esté probada. Una vez creadas las pruebas el código no pasa a producción hasta que el resultado de las mismas no sea satisfactorio. Luego se inicia el proceso de refactorización que consiste en limpiar y organizar el código, adaptarlo a los patrones y aumentar su legibilidad, todo esto sin modificar su comportamiento externo. (40)

3.7.2. Pruebas unitarias

Se encargan de probar una clase en concreto, testeando cada uno de sus métodos y viendo si dados unos parámetros de entrada, la salida es la esperada. La realización de estas pruebas según XP debe consumir la menor cantidad de tiempo posible, por lo que se recomienda el uso de herramientas. Visual Studio proporciona un sistema sencillo para poder crear pruebas unitarias que facilita la creación de los proyectos de pruebas y genera la estructura básica que tiene que tener la prueba.

3.7.3. Pruebas de aceptación

Las pruebas de aceptación son las especificaciones para el comportamiento deseado y la funcionalidad de un sistema. Expresan, por una HU dada, cómo el sistema se encarga de ciertas condiciones e insumos y con qué tipo de resultados. Los clientes junto a un miembro del equipo de desarrollo son los responsables de verificar que los resultados de estas pruebas sean los correctos para así tomar decisiones acerca de las mismas. Una HU no se puede considerar terminada hasta que no pase los *test* de aceptación. Es recomendable publicar los resultados de las pruebas de aceptación, para que todo el equipo de desarrollo esté al tanto de esta información. (40)

A continuación aparecen algunos casos de pruebas de aceptación referentes a la solución propuesta, el resto se encuentra en los anexos del documento:

Caso de prueba de aceptación	
Código: HU1_CP1	Historia de Usuario: 1
Nombre: Autenticar usuario	
Condiciones de ejecución: El usuario debe estar registrado en el sistema.	
Entrada/Pasos de ejecución: <i>Entrada:</i> La entrada consiste en los datos que el usuario debe introducir para ser autenticado en el sistema. <i>Pasos de ejecución:</i> El usuario debe llenar los campos usuario y contraseña mostrados en la ventana y debe presionar el botón aceptar. Una vez realizado el proceso, el usuario entra en el sistema.	
Resultado esperado: El usuario es autenticado correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 52. CASO DE PRUEBA DE ACEPTACIÓN #1

Caso de prueba de aceptación	
Código: HU2_CP1	Historia de Usuario: 2
Nombre: Crear Pregunta	
Condiciones de ejecución: El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura.	
Entrada/Pasos de ejecución: <i>Entrada:</i> La entrada consiste en el cuerpo de la pregunta que el usuario debe insertar, así como la selección de la asignatura, el tema, el objetivo y el nivel de complejidad. <i>Pasos de ejecución:</i> El usuario debe seleccionar en el menú el módulo preguntas, una vez en este debe presionar la opción Crear pregunta. Una vez introducido los datos de la pregunta el usuario la redacta, se presiona en el botón guardar y la pregunta es insertada en la base de datos del sistema.	
Resultado esperado: La pregunta es insertada correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 53. CASO DE PRUEBA DE ACEPTACIÓN #2

Caso de prueba de aceptación	
Código: HU2_CP2	Historia de Usuario: 2
Nombre: <i>Modificar Preguntas</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura y debe haber insertada en la base de datos alguna pregunta.</i>	
Entrada/Pasos de ejecución: <i>Entrada: La entrada consiste los datos de la pregunta que el usuario quiere modificar.</i> <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo preguntas, una vez en este debe presionar la opción Modificar pregunta. Una vez modificado los datos, presiona en el botón guardar y la pregunta es modificada en la base de datos del sistema.</i>	
Resultado esperado: <i>La pregunta es modificada correctamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 54. CASO DE PRUEBA DE ACEPTACIÓN #3

Caso de prueba de aceptación	
Código: HU2_CP3	Historia de Usuario: 2
Nombre: <i>Eliminar Preguntas</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura y debe haber insertada en la base de datos alguna pregunta.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo preguntas, una vez en este debe presionar la opción Eliminar pregunta. Realiza una búsqueda de la pregunta que desea eliminar, filtrando por asignatura, tema y objetivos. Selecciona la pregunta a eliminar, presiona en el botón eliminar y se le pregunta si está seguro que quiere eliminar. El usuario selecciona si desea o no eliminar la pregunta.</i>	
Resultado esperado: <i>La pregunta es eliminada satisfactoriamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 55. CASO DE PRUEBA DE ACEPTACIÓN #4

Caso de prueba de aceptación	
Código: HU2_CP4	Historia de Usuario: 2
Nombre: Ver Preguntas	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura y debe haber insertada en la base de datos alguna pregunta.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo preguntas, una vez en este debe presionar la opción Ver preguntas.</i>	
Resultado esperado: <i>Se mostrarán las preguntas insertadas en la base de datos por el usuario que está autenticado en el sistema.</i>	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 56. CASO DE PRUEBA DE ACEPTACIÓN #5

Caso de prueba de aceptación	
Código: HU3_CP1	Historia de Usuario: 3
Nombre: Crear exámenes manualmente	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura y debe haber insertada en la base de datos alguna pregunta.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo exámenes, una vez en este debe presionar la opción Crear exámenes, selecciona la opción manualmente y se presenta una interfaz donde el usuario selecciona las preguntas que desea para confeccionar el examen.</i>	
Resultado esperado: <i>El examen es creado satisfactoriamente.</i>	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 57. CASO DE PRUEBA DE ACEPTACIÓN #6

Caso de prueba de aceptación	
Código: HU3_CP2	Historia de Usuario: 3
Nombre: <i>Crear exámenes aleatorio</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura y debe haber insertada en la base de datos alguna pregunta.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo exámenes, una vez en este debe presionar la opción Crear exámenes, selecciona la opción aleatorio. En el nuevo visual el usuario introduce la cantidad de preguntas y para cada una de estas selecciona la complejidad, el objetivo y el tema.</i>	
Resultado esperado: <i>El examen es generado satisfactoriamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 58. CASO DE PRUEBA DE ACEPTACIÓN #7

Caso de prueba de aceptación	
Código: HU3_CP3	Historia de Usuario: 3
Nombre: <i>Modificar exámenes</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura y debe haber guardado en la base de datos algún examen.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo exámenes, una vez en este debe presionar la opción Modificar exámenes, selecciona el examen a modificar y realiza los cambios en este.</i>	
Resultado esperado: <i>El examen es modificado satisfactoriamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 59. CASO DE PRUEBA DE ACEPTACIÓN #8

Caso de prueba de aceptación	
Código: HU3_CP4	Historia de Usuario: 3
Nombre: <i>Eliminar exámenes</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura y debe haber guardado en la base de datos algún examen.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo exámenes, una vez en este debe presionar la opción Eliminar exámenes, selecciona el examen a eliminar y una vez seleccionado se le pregunta si desea o no eliminar el examen.</i>	
Resultado esperado: <i>El examen es eliminado satisfactoriamente.</i>	

Evaluación de la prueba: *Prueba satisfactoria.*

TABLA 60. CASO DE PRUEBA DE ACEPTACIÓN #9

Caso de prueba de aceptación	
Código: HU3_CP5	Historia de Usuario: 3
Nombre: <i>Ver exámenes</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura y debe haber guardado en la base de datos algún examen.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo exámenes, una vez en este debe presionar la opción Ver exámenes, selecciona el examen a visualizar.</i>	
Resultado esperado: <i>Se mostrarán los exámenes insertados en la base de datos por el usuario que está autenticado en el sistema.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 61. CASO DE PRUEBA DE ACEPTACIÓN #10

Caso de prueba de aceptación	
Código: HU3_CP6	Historia de Usuario: 3
Nombre: <i>Guardar exámenes</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo exámenes, una vez en este debe presionar la opción Crear exámenes, y una vez creado puede guardarlo en la base de datos.</i>	
Resultado esperado: <i>El examen es guardado en la base de datos satisfactoriamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 62. CASO DE PRUEBA DE ACEPTACIÓN #11

Caso de prueba de aceptación	
Código: HU4_CP1	Historia de Usuario: 4
Nombre: <i>Exportar exámenes a PDF</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol de profesor o Jefe de asignatura.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo exámenes, una vez en este debe presionar la opción Crear exámenes, y una vez creado puede exportarlo a PDF.</i>	
Resultado esperado: <i>El examen es exportado a PDF satisfactoriamente.</i>	

Evaluación de la prueba: Prueba satisfactoria.

TABLA 63. CASO DE PRUEBA DE ACEPTACIÓN #12

Caso de prueba de aceptación	
Código: HU5_CP1	Historia de Usuario: 5
Nombre: Asignar asignatura	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de asignatura o Jefe de departamento.	
Entrada/Pasos de ejecución: <i>Pasos de ejecución:</i> El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar usuario y al adicionar un usuario puede asignarle una o varias asignaturas a este.	
Resultado esperado: Le asigna la asignatura al usuario satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 64. CASO DE PRUEBA DE ACEPTACIÓN #13

Caso de prueba de aceptación	
Código: HU6_CP1	Historia de Usuario: 6
Nombre: Asignar complejidad	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de asignatura o Jefe de departamento.	
Entrada/Pasos de ejecución: <i>Pasos de ejecución:</i> Una vez autenticado el sistema muestra una notificación de las preguntas que no tienen complejidad asignada.	
Resultado esperado: Le asigna la complejidad a la pregunta satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 65. CASO DE PRUEBA DE ACEPTACIÓN #14

Caso de prueba de aceptación	
Código: HU7_CP1	Historia de Usuario: 7
Nombre: Agregar usuario	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de departamento	
Entrada/Pasos de ejecución: <i>Pasos de ejecución:</i> El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar usuario, selecciona la opción Agregar usuario y llena los datos del usuario a adicionar	

Resultado esperado: <i>El usuario es adicionado a la base de datos satisfactoriamente.</i>
Evaluación de la prueba: <i>Prueba satisfactoria.</i>

TABLA 66. CASO DE PRUEBA DE ACEPTACIÓN #15

Caso de prueba de aceptación	
Código: <i>HU7_CP2</i>	Historia de Usuario: <i>7</i>
Nombre: <i>Modificar usuario</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Jefe de departamento</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar usuario, selecciona la opción Modificar usuario y llena los datos del usuario a modificar.</i>	
Resultado esperado: <i>El usuario es modificado en la base de datos satisfactoriamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 67. CASO DE PRUEBA DE ACEPTACIÓN #16

Caso de prueba de aceptación	
Código: <i>HU7_CP3</i>	Historia de Usuario: <i>7</i>
Nombre: <i>Eliminar usuario</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Jefe de departamento</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar usuario, selecciona la opción Eliminar usuario y llena los datos del usuario a eliminar.</i>	
Resultado esperado: <i>El usuario es eliminado de la base de datos satisfactoriamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 68. CASO DE PRUEBA DE ACEPTACIÓN #17

Análisis de los resultados de las pruebas.

Al concluir cada una de las iteraciones planificadas para el desarrollo de la solución, fueron realizadas las pruebas pertinentes para realizar las entregas pactadas con el cliente, la tabla muestra el resultado de dichas pruebas en cada una de las iteraciones.

	Iteración 1	Iteración 2	Iteración 3
Cantidad	13	9	5

TABLA 69. CANTIDAD DE NO CONFORMIDADES DETECTADAS POR ITERACIÓN

Todas las no conformidades fueron resueltas logrando validar, en cierto grado, la calidad de la solución propuesta y mayor satisfacción por parte del cliente.

3.8. Conclusiones parciales

En el desarrollo del capítulo se generaron las 7 tarjetas CRC correspondientes a las clases controladoras del sistema y se determinó que la base de datos cuenta con 7 entidades como se evidencia en el Diagrama Entidad Relación. Se definió como patrón arquitectónico a utilizar el Modelo Vista Controlador y como patrones de diseño GRASP: el experto, el creador, el controlador y el bajo acoplamiento. Fueron elaboradas 26 tareas de ingenierías necesarias para desarrollar cada historia de usuario y se describieron las pruebas de aceptación realizadas, evidenciando la cantidad de no conformidades detectadas y corrigiendo estas.

CONCLUSIONES GENERALES

- Los métodos de investigación utilizados permitieron identificar los aspectos teóricos referentes a la elaboración de instrumentos evaluativos dentro del proceso docente – educativo.
- Caracterizar los sistemas existentes relacionados con la elaboración de instrumentos evaluativos, permitió identificar la necesidad de la realización de un nuevo sistema que incluyera criterio de expertos para clasificar el nivel de las preguntas.
- La metodología de *software*, las tecnologías, técnicas y herramientas seleccionadas permitieron el cumplimiento del objetivo general.
- La realización de las pruebas de aceptación demostró que el software cumple con las expectativas del cliente.

RECOMENDACIONES

Tomando como base la investigación realizada y el análisis de los resultados obtenidos se recomienda:

- Migrar el sistema a tecnologías libres para el uso de la aplicación en computadoras con sistemas operativos Linux.
- Utilizar técnicas que detecten cuando una pregunta es igual o similar a otra y notificar a los usuarios para evitar la reiteración en la base de datos del sistema.
- Desarrollar un módulo gestión de reportes que permita a los usuarios saber la cantidad de preguntas disponibles en la base de datos, la cantidad de preguntas usadas, así como el por ciento de preguntas por niveles de complejidad y asignaturas u otro criterio definido.
- Utilizar el sistema en el Departamento de Programación de la Facultad 2 para validar en tiempo real que el software optimiza el proceso de confección de exámenes en cuanto a disminución del tiempo y aumento de la calidad.

REFERENCIA BIBLIOGRÁFICA

1. FERNÁNDEZ, Inmaculada. Las TICS en el ámbito educativo. España, 2006.
2. UNESCO. Glossary of Educational Technology Terms. París, Francia, 1984.
3. MINREX. Cuba en la Cumbre Mundial sobre la Sociedad de la Información. [En línea]. 2004. [Accedido 24 marzo 2014]. Disponible en: http://anterior.cubaminrex.cu/Sociedad_Informacion/Informacion_Gral.htm.
4. AREA MOREIRA, Manuel. Introducción a la Tecnología Educativa. España, 2009.
5. CABERO, J. Nuevas tecnologías, comunicación y educación. EDUTEC. Revista Electrónica de Tecnología Educativa. [En línea]. [Accedido 28 marzo 2014]. Disponible en: <http://www.uib.es/depart/gte/revelec1.html>.
6. MALDONADO, Islas. Fundamentos de Tecnología Educativa. [En línea]. [Accedido 28 marzo 2014]. Disponible en: <http://www.americalearningmedia.com/edicion-010/122-white-papers/849-fundamentos-de-tecnologia-educativa>.
7. ÁLVAREZ ZAYAS, Carlos M. La escuela en la vida. Ciudad de la Habana, Cuba, 1999.
8. Resolución 210/2007: Reglamento trabajo docente metodológico. Ciudad de la Habana, 2007.
9. LABARRERE REYES, Guillermina, VALDIVIA PAIROL, Gladys. Pedagogía. Ciudad de la Habana, 1988.
10. VÁZQUEZ GOYANES, Sergio. Herramienta informática que permite la generación y evaluación de exámenes con múltiples variantes. Ciudad de la Habana: Universidad de las Ciencias Informáticas, 2010.
11. DEL RAMO ROMERO, José J. y LÓPEZ SOLER, Neus. Respondus, un programa para la creación y publicación de cuestionarios. [En línea]. 2008. [Accedido 28 marzo 2014]. Disponible en: ojs.uv.es/index.php/attic/article/download/53/1219
12. ANTÓN, Pastor. Herramientas de autoevaluación tipo test on-line y off-linefreeware: Hot-Potatoes y SED. : [En línea]. [Accedido 28 marzo 2014]. Disponible en: <http://www.virtualeduca.org>

13. Acerca de Moodle - MoodleDocs. [En línea]. [Accedido 28 marzo 2014]. Disponible en: http://docs.moodle.org/all/es/Acerca_de_Moodle.
14. CATHCART PORTUONDO, Eva Manuela y OTROS. Sistema generador de exámenes de medicina (GEXMEDI). Junio 2006, Vol. 5, pp. 15.
15. LUCENA LOPEZ, Manuel José. Criptografía y seguridad en computadores, 4ta edición. Escuela Politécnica Superior. Universidad de Jaén. 2005
16. PÉREZ GONZÁLEZ, Rodrigo, CARRILLO PÉREZ, Isaías y Rodríguez Martín, Aureliano David. Metodología de Desarrollo del *Software*. 2008.
17. LETELIER, Patricio, PENADÉS, Carmen y CANÓS, José. Metodologías ágiles para el desarrollo de *software*: eXtreme Programming (XP). Universidad Politécnica de Valencia. 16. PALACIOS, Juan. Scrum Manager: Gestión de proyectos, 2008.
18. HIGHSMITH, James A. Agile *Software* Development Ecosystems [En línea]. Addison-Wesley Professional, 2002. ISBN 9780201760439. Disponible en: http://www.google.com/cu/books?id=uE4FGFOHs2EC&lpg=PP17&ots=OL0loD3o4q&dq=Agile%20software%20development%20ecosystems.%20&lr=lang_es%7Clang_en&hl=es&pg=PP17#v=onepage&q=Agile%20software%20development%20ecosystems.&f=false.
19. WILSON, Leslie Blackett. Comparative Programming Languages. 1993. ISBN 0-201-56885-3.
20. Microsoft. Introducción al lenguaje C# y .NET Framework. [En línea]. [Accedido 24 abril 2014]. Disponible en: <http://msdn.microsoft.com/es-es/library/z1zx9t92.aspx>.
21. PROGRAMACIÓN JAVA: Entorno de desarrollo integrado (IDE). [En línea] [Accedido 24 abril 2014]. Disponible en: <http://programacion-laura.blogspot.com>
22. ABREGO, Maleny. ¿Visual Estudio? [En línea]. [Accedido 28 abril 2014]. Disponible en: <http://malenyabrego.wordpress.com/2011/11/10/%C2%BFvisual-estudio/>.
23. BRUNIS, Christian. Visual Studio 2010. [En línea]. [Accedido 28 abril 2014]. Disponible en: <http://blog.espol.edu.ec/doucapol/2012/01/03/visual-studio-2010/>.

24. BERTINO, E. y MARTINO, L. Sistemas de bases de datos orientadas a objetos. Ediciones Díaz de Santos, 1995.
25. Microsoft. Microsoft SQL Server 2008 R2 [En línea]. [Accedido 29 abril 2014]. Disponible en: <http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>
26. Unified Modeling Language (UML). IBM [En línea] [Accedido 24 abril 2014]. Disponible en: <http://www.ibm.com/developerworks/rational/library/>
27. MCCLURE, Carma. The CASE Experience. Abril 1989.
28. Visual Paradigm. [En línea] [Accedido 24 abril 2014]. Disponible en: <http://www.visual-paradigm.com/>.
29. BAEZA, Pablo Nicolás. Visual Paradigm DB Visual ARCHITECT SQL. [En línea] [Accedido 24 abril 2014]. Disponible en: <http://www.docstoc.com/docs/96492173/Visual-Paradigm-Studio>.
30. CALABRIA, Luis; PÍRIZ, Pablo. Metodología XP. Universidad ORT Uruguay: Cátedra de Ingeniería de *Software*. 2003.
31. JEFFRIES, Ron, ANDERSON, Ann y HENDRICKSON, Chet. Extreme Programming Installed [En línea] Addison-Wesley Professional, 2001. ISBN 9780201708424. Disponible en: http://www.google.com/cu/books?id=5ZuPjdO8LLoC&lpg=PR13&ots=OHXBtdUX_H&dq=Extreme%20Programming%20Explained.%20Embrace%20Change&lr&pg=PR13#v=onepage&q=Extreme%20Programming%20Explained.%20Embrace%20Change&f=false.
32. BECK, Kent. Extreme Programming Explained, 1999.
33. CARVAJAL RIOLA, José Carlos. Metodologías ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial. 2008.
34. BECK, K. y CUNNINGHAM, W. A Laboratory for Teaching Object Oriented Thinking. Conference Proceedings on Object-oriented Programming Systems, Languages and Applications [En línea] New York, NY, USA: ACM, 1989. p. 1–6. [Accedido 1 abril 2014]. ISBN 0-89791-333-7. Disponible en: <http://doi.acm.org/10.1145/74877.74879>.
35. LARMAN, C. UML y patrones. Tomo I Capítulos 18, Páginas 185-215.

-
36. PBworks. Patrones arquitectónicos. [En línea]. [Accedido 29 abril 2014]. Disponible en: <http://isg3.pbworks.com/w/page/7624479/Patrones%20Arquitect%C3%B3nicos>.
37. HENNEY, Kevlin. ¿What is *Software Architecture*? 2007.
38. GAMMA, E., HELM, R., JOHNSON, R. y VLISSIDES, J. Patrones de diseño. 2000.
39. RODRIGUEZ CORBEA, Maite, ORDONEZ PEREZ, Meylin coautora and PEREZ HEREDIA, Yaneisis González. La Metodología XP Aplicable al Desarrollo del *Software* educativo en Cuba. [En línea]. 2007. [Accedido 24 April 2014]. Disponible en: http://repositorio_institucional.uci.cu//jspui/handle/ident/TD_0837_07.
40. KOSKELA, Lasse. TEST DRIVEN, Practical TDD and Acceptance TDD for Java Developers. Manning Publication Co., 2008. ISBN 1-932394-85-0.

BIBLIOGRAFÍA

Acerca de Moodle - MoodleDocs. [En línea]. [Accedido 28 marzo 2014]. Disponible en: http://docs.moodle.org/all/es/Acerca_de_Moodle.

ÁLVAREZ ZAYAS, Carlos M. La escuela en la vida, Ciudad de la Habana, 1999.

ÁLVAREZ, Zenaida. La evaluación como método de educación. ISPETP. Cuba, 2001.

AmericaTI. Ventajas y Desventajas: Comparación de los Lenguajes C, C++ y Java. [En línea]. 2006. [Accedido 24 abril 2014]. Disponible en: http://www.americati.com/doc/ventajas_c.pdf.

ANTÓN, Pastor. Herramientas de autoevaluación tipo test on-line y off-linefreeware: Hot-Potatoes y SED. [En línea]. [Accedido 28 marzo 2014]. Disponible en: <http://www.virtualeduca.org>

AREA MOREIRA, Manuel. Introducción a la Tecnología Educativa. España, 2009.

BAEZA, Pablo Nicolás. Visual Paradigm DB Visual ARCHITECT SQL. [En línea]. [Accedido 24 abril 2014]. Disponible en: <http://www.docstoc.com/docs/96492173/Visual-Paradigm-Studio>.

BERTINO, E. A. y MARTINO, L. A. Sistemas de bases de datos orientadas a objetos. Ediciones Díaz de Santos, 1995.

BRUNIS, Christian. Visual Studio 2010. [En línea]. [Accedido 28 abril 2014]. Disponible en: <http://blog.espol.edu.ec/doucapol/2012/01/03/visual-studio-2010/>.

BECK, Kent. Extreme Programming Explained, 1999.

BECK, K. y CUNNINGHAM, W. A Laboratory for Teaching Object Oriented Thinking. Conference Proceedings on Object-oriented Programming Systems, Languages and Applications [En línea] New York, NY, USA: ACM, 1989. p. 1–6. [Accedido 1 abril 2014]. ISBN 0-89791-333-7. Disponible en: <http://doi.acm.org/10.1145/74877.74879>.

CABERO, J. Nuevas tecnologías, comunicación y educación. EDUtec. Revista Electrónica de Tecnología Educativa. [En línea]. [Accedido 28 marzo 2014]. Disponible en: <http://www.uib.es/depart/gte/revelec1.html>.

CALABRIA, Luis; PÍRIZ, Pablo. Metodología XP. Universidad ORT Uruguay: Cátedra de Ingeniería de Software., 2003.

CARVAJAL RIOLA, José Carlos. Metodologías ágiles: Herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial. 2008.

CASTRO, Pimienta. Evaluación en la escuela actual ¿Reduccionismo o desarrollo? CEPROF. ISPETP. Cuba, 1997.

CATHCART PORTUONDO, Eva Manuela y OTROS. Sistema generador de exámenes de medicina (GEXMEDI). Junio 2006, Vol. 5, pp. 15.

Cuba. Ministerio de la Informática y las Comunicaciones. Informatización de la Sociedad. [En línea] [Accedido 24 marzo 2013]. Disponible en: <http://www.mic.gov.cu/sitiomic/hinfosoc.asp>

DE NOBREGA, María. Herramientas CASE: Rational Rose. [En línea]. [Accedido 24 abril 2014]. Disponible en: http://curso_sin2.blogia.com/2005/060401-herramientas-case-rational-rose.-por-maria-de-nobrega.php.

DEL RAMO ROMERO, José J. y LÓPEZ SOLER, Neus. Respondus, un programa para la creación y publicación de cuestionarios. [En línea]. 2008. [Accedido 28 marzo 2014]. Disponible en: ojs.uv.es/index.php/attic/article/download/53/1219

DELGADO, Vera y PALACIOS, Rafael. Introducción a la Criptografía: tipos de algoritmos, 2006.

Dugarte, Ana, y otros. 2009. Lenguaje Unificado de Modelado. [En línea] 2009. [Accedido 28 abril 2014]. Disponible en: <http://www.oocities.org/es/annadugarte/ads1/PRINCIPAL.htm>

GAMMA, E., HELM, R., JOHNSON, R. y VLISSIDES, J. Patrones de diseño. 2000.

FERNÁNDEZ, Inmaculada. Las TICS en el ámbito educativo. España, 2006.

HIGHSMITH, James A. Agile *Software* Development Ecosystems [En línea]. Addison-Wesley Professional, 2002. ISBN 9780201760439. Disponible en: http://www.google.com/cu/books?id=uE4FGFOHs2EC&lpg=PP17&ots=OL0loD3o4q&dq=Agile%20software%20development%20ecosystems.%20&lr=lang_es%7Clang_en&hl=es&pg=PP17#v=onepage&q=Agile%20software%20development%20ecosystems.&f=false.

JEFFRIES, Ron, ANDERSON, Ann y HENDRICKSON, Chet. Extreme Programming Installed [En línea]. Addison-Wesley Professional, 2001. ISBN 9780201708424. Disponible en: http://www.google.com/cu/books?id=5ZuPjdO8LLoC&lpg=PR13&ots=OHXBtdUX_H&dq=Extreme%20Programming%20Explained.%20Embrace%20Change&lr&pg=PR13#v=onepage&q=Extreme%20Programming%20Explained.%20Embrace%20Change&f=false.

JOSKOWICZ, José. Reglas y Prácticas en eXtreme Programming, 2008.

KOSKELA, Lasse. TEST DRIVEN, Practical TDD and Acceptance TDD for Java Developers. Manning Publication Co., 2008. ISBN 1-932394-85-0.

LARMAN, C. UML y patrones. Tomo I Capítulos 18, Páginas 185-215.

LETELIER, Patricio, PENADÉS, Carmen y CANÓS, José. Metodologías ágiles para el desarrollo de *software*: eXtreme Programming (XP). Universidad Politécnica de Valencia. LUCENA LOPEZ, Manuel José. Criptografía y seguridad en computadores, 4ta edición. Escuela Politécnica Superior. Universidad de Jaén. 2005

MALDONADO, Islas. Fundamentos de Tecnología Educativa. [En línea]. [Accedido 28 marzo 2014]. Disponible en: <http://www.americalearningmedia.com/edicion-010/122-white-papers/849-fundamentos-de-tecnologia-educativa>.

MCCLURE, Carma. The CASE Experience. Abril 1989.

Microsoft. Introducción al lenguaje C# y .NET Framework. [En línea]. [Accedido 24 abril 2014]. Disponible en: <http://msdn.microsoft.com/es-es/library/z1zx9t92.aspx>.

Microsoft. Microsoft SQL Server 2008 R2 [En línea]. [Accedido 29 abril 2014]. Disponible en: <http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>

MINISTERIO DE EDUCACIÓN SUPERIOR. Resolución 210: Reglamento trabajo docente metodológico. Ciudad de la Habana, 2006.

MINREX. Cuba en la Cumbre Mundial sobre la Sociedad de la Información. [En línea]. 2004. [Accedido 24 marzo 2014]. Disponible en: http://anterior.cubaminrex.cu/Sociedad_Informacion/Informacion_Gral.htm.

Patrones de diseño. [En línea]. [Accedido 29 abril 2014]. Disponible en: <http://sourcemaking.com/patr%C3%B3n-de-dise%C3%B1o>

PBworks. Patrones arquitectónicos. [En línea]. [Accedido 29 abril 2014]. Disponible en: <http://isg3.pbworks.com/w/page/7624479/Patrones%20Arquitect%C3%B3nicos>.

PÉREZ GONZÁLEZ, Rodrigo, CARRILLO PÉREZ, Isaías y Rodríguez Martín, Aureliano David. Metodología de Desarrollo del *Software*. 2008.

Portal de la Universidad de las Ciencias Informáticas. [En línea]. 2004. [Accedido 24 marzo 2014]. Disponible en: <http://www.uci.cu/?q=mision>.

Procesos de *Software* Metodología Extreme Programming (XP). [En línea]. 2004. [Accedido 1 abril 2014]. Disponible en: <http://procesosdesoftware.wikispaces.com/METODOLOGIA+XP>

RODRIGUEZ CORBEA, Maite, ORDONEZ PEREZ, Meylin, PEREZ HEREDIA, Yaneisis González. La Metodología XP Aplicable al Desarrollo del *Software* educativo en Cuba. [En línea]. 2007. [Accedido 24 April 2014]. Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_0837_07.

UNESCO. Glossary of Educational Technology Terms. París, Francia, 1984.

Unified Modeling Language (UML). IBM. [En línea] [Accedido 24 abril 2014]. Disponible en: <http://www.ibm.com/developerworks/rational/library/>

VÁZQUEZ GOYANES, Sergio. Herramienta informática que permite la generación y evaluación de exámenes con múltiples variantes. Ciudad de la Habana: Universidad de las Ciencias Informáticas, 2010.

Visual Paradigm. [En línea]. [Accedido 24 abril 2014]. Disponible en: <http://www.visual-paradigm.com/>

WILSON, Leslie Blackett. Comparative Programming Languages. 1993. ISBN 0-201-56885-3.

ANEXOS

Guía para la entrevista al jefe del Departamento de Programación

1. Cantidad de profesores del departamento de programación.
2. Cantidad de asignaturas que se imparten.
3. Categorías docentes de los profesores del departamento.
4. Aspecto que se tienen en cuenta en el proceso de confección de los instrumentos evaluativos.
5. Principales deficiencias del proceso de confección de los instrumentos evaluativos.

Guía para la entrevista a profesores del Departamento de Programación

1. Principales problemas en la confección de preguntas para los instrumentos evaluativos.
2. Preparación pedagógica que posee.

Caso de prueba de aceptación	
Código: HU9_CP3	Historia de Usuario: 9
Nombre: Eliminar tema	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de asignatura.	
Entrada/Pasos de ejecución: Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar tema, selecciona la opción Eliminar tema y selecciona si desea o no eliminar el tema.	
Resultado esperado: El tema es eliminado de la base de datos satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 70. CASO DE PRUEBA DE ACEPTACIÓN #23

Caso de prueba de aceptación	
Código: HU10_CP1	Historia de Usuario: 10
Nombre: Agregar objetivo	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de asignatura.	
Entrada/Pasos de ejecución: Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar objetivo, selecciona la opción Agregar objetivo y llena los datos del objetivo a agregar.	
Resultado esperado: El objetivo es agregado a la base de datos satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 71. CASO DE PRUEBA DE ACEPTACIÓN #24

Caso de prueba de aceptación	
Código: HU10_CP2	Historia de Usuario: 10
Nombre: Editar objetivo	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de asignatura.	
Entrada/Pasos de ejecución: Pasos: El usuario debe seleccionar en el menú el módulo configuración, una vez en este la opción Gestionar objetivo, selecciona la opción Editar objetivo y llena los datos del objetivo a editar.	
Resultado esperado: El objetivo es editado de la base de datos satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 72. CASO DE PRUEBA DE ACEPTACIÓN #25

Caso de prueba de aceptación	
Código: HU10_CP3	Historia de Usuario: 10
Nombre: Eliminar objetivo	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de asignatura.	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar objetivo, selecciona la opción Eliminar objetivo y selecciona si desea o no eliminar el objetivo.</i>	
Resultado esperado: El objetivo es eliminado de la base de datos satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 73. CASO DE PRUEBA DE ACEPTACIÓN #26

Caso de prueba de aceptación	
Código: HU8_CP1	Historia de Usuario: 8
Nombre: Adicionar asignatura	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de departamento	
Entrada/Pasos de ejecución: Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar asignatura, selecciona la opción Agregar asignatura y llena los datos del asignatura a adicionar.	
Resultado esperado: La asignatura es adicionada a la base de datos satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 74. CASO DE PRUEBA DE ACEPTACIÓN #18

Caso de prueba de aceptación	
Código: HU8_CP2	Historia de Usuario: 8
Nombre: Modificar asignatura	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de departamento	
Entrada/Pasos de ejecución: Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar asignatura, selecciona la opción Modificar asignatura y llena los datos del asignatura a adicionar.	
Resultado esperado: La asignatura es modificada en la base de datos satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

TABLA 75. CASO DE PRUEBA DE ACEPTACIÓN #19

Caso de prueba de aceptación	
Código: HU8_CP3	Historia de Usuario: 8
Nombre: Eliminar asignatura	
Condiciones de ejecución: El usuario debe estar autenticado con el rol Jefe de departamento	
Entrada/Pasos de ejecución: Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar asignatura, selecciona la opción Eliminar asignatura y selecciona si desea o no eliminar la asignatura.	
Resultado esperado: La asignatura es eliminada de la base de datos satisfactoriamente.	

Evaluación de la prueba: *Prueba satisfactoria.*

TABLA 76. CASO DE PRUEBA DE ACEPTACIÓN #20

Caso de prueba de aceptación	
Código: HU9_CP1	Historia de Usuario: 9
Nombre: <i>Agregar tema</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Jefe de asignatura.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar tema, selecciona la opción Agregar tema y llena los datos del tema a adicionar.</i>	
Resultado esperado: <i>El tema es adicionado a la base de datos satisfactoriamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 77. CASO DE PRUEBA DE ACEPTACIÓN #21

Caso de prueba de aceptación	
Código: HU9_CP2	Historia de Usuario: 9
Nombre: <i>Editar tema</i>	
Condiciones de ejecución: <i>El usuario debe estar autenticado con el rol Jefe de asignatura.</i>	
Entrada/Pasos de ejecución: <i>Pasos de ejecución: El usuario debe seleccionar en el menú el módulo configuración, una vez en este debe presionar la opción Gestionar tema, selecciona la opción Editar tema y llena los datos del tema a editar.</i>	
Resultado esperado: <i>El tema es modificado en la base de datos satisfactoriamente.</i>	
Evaluación de la prueba: <i>Prueba satisfactoria.</i>	

TABLA 78. CASO DE PRUEBA DE ACEPTACIÓN #22

GLOSARIO DE TÉRMINOS

.NET: Es un framework de Microsoft que hace énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones.

Borland Delphi: Es un entorno de desarrollo flexible y potente. Es intérprete de un lenguaje llamado Object Pascal, incluye herramientas para facilitar la escritura del código y el diseño de la aplicación.

CASE: (Computer Aided *software* Engineering, Ingeniería de *Software* Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software*. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del *software* en tareas como el proceso de realizar un diseño del proyecto.

E-learning: es la simplificación de Electronic Learning. El mismo reúne a las diferentes tecnologías, y a los aspectos pedagógicos de la enseñanza y el aprendizaje.

IBM: International Business Machines es una empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk, Nueva York. IBM fabrica y comercializa hardware y *software* para computadoras, y ofrece servicios de infraestructura, alojamiento de Internet, y consultoría en una amplia gama de áreas relacionadas con la informática, desde computadoras centrales hasta nanotecnología.

ISO: es la Organización Internacional de Normalización, nacida tras la Segunda Guerra Mundial (23 de febrero de 1947), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación (tanto de productos como de servicios), comercio y comunicación para todas las ramas industriales. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.

PDF: Es un formato de almacenamiento de documentos digitales independiente de plataformas de *software* o hardware. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto). Fue inicialmente desarrollado por la empresa Adobe Systems, oficialmente lanzado como un estándar abierto el 1 de julio de 2008 y publicado por la Organización Internacional de Estandarización como ISO 32000-1.

PIN (Personal Identification Number) es un número de identificación personal utilizado en ciertos sistemas, como el teléfono móvil o el cajero automático, para identificarse y obtener acceso al sistema. El PIN es un tipo de contraseña. Sólo la persona beneficiaria del servicio conoce el PIN que le da acceso al mismo; esa es su finalidad. El PIN tiene que ser suficientemente seguro evitar la intrusión no autorizada al servicio que protege.

RAM: (random-access memory) se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del *software*. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo. Se denominan de acceso aleatorio porque se puede leer o escribir en una posición de memoria con un tiempo de espera igual para cualquier posición, no siendo necesario seguir un orden para acceder a la información de la manera más rápida posible.