



TESIS EN OPCIÓN AL TÍTULO DE MASTER

Título: Propuesta de curso de metodología ágil. Scrum.

Autor: Lic. Liset López González.

Tutores: MsC. Dailyn Sosa López.

MsC. Kadir Hector Ortiz.

Curso 2010-2011

“El problema no es el cambio, porque el cambio, inevitablemente, va a ocurrir.

El problema, realmente, es la inhabilidad para hacer frente a los cambios”.

Kent Beck,

A mis padres
A mis hijas, mi razón de ser.
A ti pequeña Camila.

A mi hija Daifyn, porque se lo debía, voz de aliento en este sueño.

A Daniela, por creer en mi.

A mi esposo, compañero de viaje.

A Bety, por sus sugerencias oportunas.

A Kadir, por su incondicionalidad.

Y a todos los que robaron tiempo a sus atareados días para ayudarme.



UNIVERSIDAD DE CIEFUEGOS
"CARLOS RAFAEL RODRÍGUEZ"

Hago constar que el presente trabajo fue realizado en la Universidad de Cienfuegos como parte de la culminación de la Maestría "TIC para la Educación", autorizando a que el mismo sea utilizado por la institución para los fines que estime conveniente, tanto en forma parcial como total, y que además no podrá ser presentado en eventos ni publicado sin la aprobación de la institución.

Nombre y Apellidos del Autor: Liset López Gonzáles

Firma_____

Los abajo firmantes certificamos que el presente trabajo ha sido revisado y el mismo cumple los requisitos establecidos, referidos a la temática señalada.

Información Científico - Técnica

Nombre y Apellidos

Tutor: Msc. Dailyn Sosa López

Firma:

Tutor: Msc. Kadir Hector Ortiz

Firma:

Resumen

Esta investigación tiene como objetivo la propuesta de un curso de metodologías ágiles de desarrollo de software, en particular SCRUM, para los estudiantes de la carrera de Ingeniería Informática de la Universidad de Cienfuegos, mediante un diseño instruccional montado en la plataforma Moodle. Los estudios realizados hasta el momento sobre el tema evidencian que estas se han tratado como ciencia individual, pero no se le ha dado un enfoque pedagógico para integrarla al proceso de enseñanza aprendizaje del ingeniero informático.

Desde el punto de vista teórico aporta el marco conceptual de las metodologías ágiles. Como significación práctica permite al estudiante apropiarse de una metodología moderna de análisis y diseño de software y aplicarla en la ejecución de las etapas tempranas de un proyecto informático. Su actualidad radica en que posibilita el vínculo entre los acelerados avances científico-técnico en relación a la producción de softwares con el proceso de enseñanza del futuro ingeniero informático.

Índice

Introducción.....	1
Capítulo 1: Fundamentación Teórica.....	7
I.1 Proceso de enseñanza aprendizaje.....	7
I.2 Enseñanza asistida por computadora (EAC)	9
I.3 Las Tecnologías de la Informática y las Comunicaciones (TIC) en la educación.	10
I.4 Modalidades de estudio, Presencial y semipresencial.....	11
I.5 Plataformas existentes en los entornos educativos	12
I.6 Ingeniería de Software.....	14
I.7 Metodologías de Desarrollo de Software	15
I.7.1 Metodologías Tradicionales.....	16
I.7.2 Metodologías Ágiles	18
I.8 Caracterización de la carrera Ingeniería Informática.....	25
I.9 Determinación de necesidades del estudiante de la carrera Ingeniería Informática para el análisis y diseño de software.	26
I.10 Módulo Instruccional	29
I.10.1 El Diseño Instruccional	29
I.10.2 Fases del Diseño Instruccional	30
I.10.3 Descripción del Módulo instruccional. Fundamentación	31
Conclusiones	34
Capítulo 2: Fundamentación y descripción de la solución propuesta.....	35
2.1 Fundamentación de la solución propuesta.	35
2.2 Pasos en la elaboración de la propuesta:	37
2.3 Propuesta del Módulo Instruccional	38
2.3.1 Estructura del Curso.....	38
2.3.2 Información General del Curso	39
2.4 Implementación del curso en Moodle	49
2.4.1 Estructura en módulos de Moodle.....	50
2.4.2 Seguridad en Moodle	52
2.4.3 Interfaz de Usuario	52

Conclusiones del capítulo.....	53
Capítulo 3: “Validación de la Propuesta”	54
3.1 Criterios de validez de la propuesta del curso de Metodologías Ágiles con Scrum.	54
3.2 Selección de los participantes en la valoración.	55
Conclusiones del capítulo.....	59
Conclusiones.....	60
Recomendaciones.....	61
Bibliografía	62
Anexos	66

Índice de Tablas

Tabla 1.	: Diferencias entre Metodologías Tradicionales y Ágiles.	24
Tabla 2.	Estructura del Curso.....	38
Tabla 3.	Ubicación Curricular.	39
Tabla 4.	Contenidos y temarios.....	44
Tabla 5.	Plan del Curso.....	46
Tabla 6.	Momentos del Proceso.....	55

Índice de Figuras

Figura 1. Fases del Diseño Instruccional según el Modelo ADDIE. Fuente: (González, 2006).....	30
Figura 2. : Secuencia del proceso de validacion de la propuesta.....	56

Introducción

Las actuales características de la industria del software han precisado replantear los cimientos sobre los que se sustenta el desarrollo de software convencional. Un estudio realizado por (Boehm, 2010), sobre la tendencia en ingeniería del software, indica que el mercado actual está caracterizado por el rápido desarrollo de aplicaciones y la reducción de la vida de los productos. En este entorno inestable la ventaja competitiva se encuentra en aumentar la productividad y satisfacer las variantes necesidades del cliente en el menor tiempo posible para proporcionar un mayor valor al negocio. Ante esta situación, cabe reflexionar sobre el grado de adaptación de las metodologías convencionales a estas circunstancias.

La mayoría de los estudios coinciden en que el carácter normativo y la fuerte dependencia de planificaciones previas al desarrollo que definen a las metodologías convencionales, implican que resulten excesivamente *pesadas* para cubrir las necesidades de un amplio porcentaje del mercado de software actual.

Desde fines de la década pasada, las metodologías ágiles han propuesto un nuevo paradigma para el desarrollo de software, que establece la incertidumbre y el consecuente cambio como realidades intrínsecas a esta actividad, tanto en lo que se refiera a los problemas a resolver como en las formas de solucionarlos. Según (Beck, 2004), uno de los padres del modelo ágil, “cada cosa en software cambia”, los requisitos cambian, el diseño cambia, el negocio cambia, la tecnología cambia, el equipo cambia y los miembros del equipo cambian. El problema no es el cambio, porque el cambio, inevitablemente, va a ocurrir. El problema, realmente, es la inhabilidad para hacer frente a los cambios”. Evidentemente, se puede concluir que parte del mercado software está variando y que es necesario abordar este cambio con nuevas alternativas que permitan adaptarse a esta situación.

En este contexto las metodologías ágiles aparecen como una opción atractiva pues, en contraposición con las metodologías convencionales que actúan basadas en principios de estabilidad y control del contexto, las metodologías ágiles no se

centran en habilidades de predicción, ni pretenden tener un sistema perfectamente definido como paso previo a su construcción, sino que perciben cada respuesta al cambio como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente, considerando la gestión de cambios como un aspecto inherente al propio proceso de desarrollo de software y, permitiendo de este modo, una mejor adaptación en entornos turbulentos.

Por otro lado, un porcentaje muy alto del desarrollo de software actual se centra en la evolución de productos ya construidos, con el objetivo de maximizar la inversión y reducir los costos, ya sea para variar parte de su funcionalidad, añadir otra nueva, o por ejemplo, adaptar el sistema a un nuevo dominio de aplicación. De hecho, algunos estudios indican que la evolución o mantenimiento del software consume un 80% del presupuesto, mientras que el nuevo desarrollo comprende el 20% restante. Este tipo de desarrollo software tiene sus propias particularidades y los nuevos enfoques que se planteen deben considerar también esta circunstancia.

El término ágil comienza a emplearse para caracterizar metodologías luego de una reunión celebrada en Utah-EEUU en febrero de 2001, se dice que fue en esa reunión donde un grupo de 17 expertos esbozaron los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Pero según otro artículo publicado por (Orjuela y Rojas, 2008) titulado “Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería de Software Educativo” fue realizada otra reunión en marzo de 2001, la cual fue convocada por (Beck, 2010) donde se acuñó el término “Métodos Ágiles” para definir a los métodos que estaban surgiendo como alternativa a las metodologías formales a las que se consideraba “pesada” y rígida por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo. Los integrantes de esta reunión resumieron los principios sobre los que se basan los métodos alternativos en cuatro postulados, lo que ha quedado denominado como Manifiesto Ágil. Plantea además este artículo que las metodologías de desarrollo ágil son más orientadas a procesos de pocas semanas y con bajos niveles de formalización en la

documentación requerida, es decir, proyectos pequeños.

A pesar del enorme acogimiento que estas metodologías están teniendo en el mercado, aún se encuentran en una etapa de crecimiento y se concluye la necesidad de realizar más investigaciones que permitan su consolidación, principalmente en el área de evolución de productos, integración continua y en la enseñanza.

Ante el imperativo de incluir las metodologías de desarrollo de software en el proceso de enseñanza aprendizaje en la formación del ingeniero informático, se aplicaron técnicas de investigación donde se hizo evidente que existen insuficiencias en el tratamiento metodológico de las metodologías ágiles de forma integrada, por la falta de preparación teórico metodológico del profesor, su preparación tradicional especializada en metodologías convencionales de desarrollo de software, y que solo la disciplina Ingeniería y Gestión de Software toma como referencia la metodología tradicional Rational Unified Process (RUP). De la situación antes descrita se define como **problema a resolver**: la necesidad de incluir contenidos relacionados a la aplicación de metodologías de desarrollo de Software Ágil, en particular SCRUM, en la formación del Ingeniero Informático.

Idea a Defender

Una propuesta de curso de metodología ágil para el desarrollo de software, en particular SCRUM, mediante un diseño instruccional montado en la plataforma Moodle, contribuirá al perfeccionamiento del proceso de enseñanza-aprendizaje en la carrera de Ingeniería Informática de la Universidad de Cienfuegos.

Objeto de estudio

Proceso de enseñanza aprendizaje de los estudiantes de cuarto año de la carrera de Ingeniería Informática

Campo de Acción

Metodologías Ágiles de desarrollo de software con SCRUM.

Objetivo General

Elaborar una propuesta de curso de metodologías ágiles de desarrollo de software, en particular SCRUM , mediante un diseño instruccional montado en la plataforma Moodle, que permita a los estudiantes de la carrera de Ingeniería Informática apropiarse de conocimientos en el desarrollo de productos informáticos.

Tareas Científicas.

- 1- Diagnóstico de las necesidades relacionadas con la preparación de los estudiantes de la carrera de Ingeniería Informática para la aplicación de metodologías ágiles de desarrollo de software.
- 2- Conceptualización teórico metodológica sobre el uso de las Tecnologías de informática y las Comunicaciones (TIC) en el proceso de enseñanza aprendizaje de la carrera ingeniería informática y las metodologías ágiles de desarrollo de software.
- 3- Diseño del curso “Metodologías Ágiles con Scrum”
- 4- Validación de la propuesta del curso.

La relación de lo empírico y lo teórico caracterizan la metodología cualitativa empleada, sustentada en el uso de técnicas como: entrevistas (Anexo 1,2) para conocer cómo se le da tratamiento a la aplicación de las metodologías ágiles en el proceso de enseñanza aprendizaje de la carrera Ingeniería Informática; el análisis de documentos relacionado con el estudio de los documentos normativos que rigen la carrera, planes de estudio, planes de clases y modelo del profesional (Anexo 3); observación en aquellas acciones que están dirigidas fundamentalmente al análisis y diseño de productos informáticos que son evaluados en clases prácticas, proyectos de curso, prácticas pre-profesional.

Se aplicaron métodos del nivel teórico tales como el analítico-sintético, para la confección de instrumentos, elaboración de la información e interpretación de diferentes fuentes, determinación de lo esencial en cada uno de ellos para arribar

a la propuesta de solución, la inducción-deducción, a partir de una muestra, arribar a conclusiones generales acerca de la preparación del estudiante en el uso de metodologías ágiles, histórico-lógico, para relacionar los antecedentes del tema, las exigencias de la carrera y la propuesta de curso.

La población está compuesta por los estudiantes de la carrera de Ingeniería Informática. Para el estudio se eligió como muestra al cuarto año de la misma.

La propuesta como **significación práctica** ofrece a los estudiantes de la carrera de Ingeniería Informática una propuesta de un curso para prepararlos en la aplicación de una metodología moderna de análisis y diseño de software y aplicarla en la ejecución de las etapas tempranas de un proyecto informático, desde el punto de vista teórico **aporta** el marco conceptual de las metodologías ágiles, en particular Scrum.

Su actualidad radica en que posibilita el vínculo entre los acelerados avances científico-técnico en relación a la producción de softwares con el proceso de enseñanza del futuro ingeniero informático.

El trabajo se adecuó de forma tal que el orden propiciara un entendimiento del argumento teórico, cumpliendo con la metodología de la investigación y facilitando el cumplimiento de los objetivos propuestos para el proyecto, por lo que su estructuración es de la manera siguiente: tres capítulos, conclusiones, recomendaciones, bibliografías y anexos.

Capítulo I: Fundamentación teórica. En este capítulo se hace un estudio del estado del arte, analizando bibliografías sobre el desarrollo del proceso de enseñanza aprendizaje, sobre el uso de las computadoras en este, enfatizando en el impacto de las nuevas tecnologías en los estudiantes así como la necesidad del diseño de una propuesta de curso con contenido novedoso y necesario para Ingenieros Informáticos o cualquier estudiante interesado. Se analizan las ventajas y desventajas de metodologías de desarrollo de software tanto tradicional como ágil.

Capítulo II: Descripción de la Solución Propuesta. En este capítulo se describen los aspectos relacionados con la propuesta de curso, explicando la necesidad del mismo y detallando las secciones que componen el modelo instruccional.

Capítulo III: Validación de la propuesta: En este capítulo se realiza la validación de la propuesta del curso de acuerdo al criterio de especialistas.

Capítulo 1: Fundamentación Teórica

En este capítulo se hace un estudio del estado del arte, analizando bibliografías sobre el desarrollo del proceso de enseñanza aprendizaje, sobre el uso de las computadoras en el proceso docente educativo, enfatizando en el impacto de las nuevas tecnologías en los estudiantes así como la necesidad del diseño de una propuesta de curso con contenido novedoso y necesario para ingenieros informáticos o cualquier estudiante interesado, con los contenidos relacionados a la aplicación de metodologías de desarrollo de software ágil.

I.1 Proceso de enseñanza aprendizaje

Como proceso de enseñanza - aprendizaje se define "el movimiento de la actividad cognoscitiva de los alumnos bajo la dirección del maestro, hacia el dominio de los conocimientos, las habilidades, los hábitos y la formación de una concepción científica del mundo" (Bartolomé, 1995). Se considera que en este proceso existe una relación dialéctica entre profesor y estudiante, los cuales se diferencian por sus funciones; el profesor debe estimular, dirigir y controlar el aprendizaje de manera tal que el alumno sea participante activo, consciente en dicho proceso, o sea, "enseñar" y la actividad del alumno es "aprender".

La relación maestro - alumno ocupa un lugar fundamental en este contexto del proceso de enseñanza-aprendizaje; el maestro tiene una función importante y los medios de enseñanza multiplican las posibilidades de ejercer una acción más eficaz sobre los alumnos.

Varios son los investigadores que se han dedicado a profundizar en el estudio de la teoría sobre los medios de enseñanza, su definición, metodología de aplicación, influencias en el proceso de asimilación de conocimientos y desarrollo de habilidades, entre otros temas.

La definición de medios de enseñanza ha sido abordada por diferentes autores; por ejemplo, Lothar Klingberg los concibe como ".todos los medios materiales

necesarios para el maestro o el alumno, para una estructuración y conducción efectiva y racional del proceso de educación e instrucción."

El colectivo de autores (Gómez, 2007) define "...los medios de enseñanza constituyen distintas imágenes y representaciones de objetos y fenómenos que se confeccionan especialmente para la docencia.", este concepto enmarca los medios de percepción directa, imágenes fijas y audiovisuales, excluyendo otros como los medios de laboratorio, equipos sonoros, las computadoras, entre otros.

El autor Vicente González Castro (Torres, 2007) define: ".Los medios de enseñanza son todos aquellos componentes del proceso docente - educativo que le sirven de soporte material a los métodos de enseñanza para posibilitar el logro de los objetivos planteados". En esta definición se reconoce como medios de enseñanza tanto a los medios visuales y sonoros como a los objetos reales, a los libros de textos, los laboratorios y a todos los recursos materiales que sirven de sustento al trabajo del maestro.

Dentro de los múltiples medios de enseñanza que se emplean en la actualidad, resulta de especial interés para este trabajo la computadora, principalmente su uso e influencia sobre el proceso de enseñanza - aprendizaje.

En las tesis de García de la Vega (García y Grau, 2007) y Pérez Fernández V. (Pérez y Díaz, 2007), que expresa ".Las computadoras son máquinas de enseñar universales, que permiten realizar a un nivel cualitativamente superior las funciones de todos los medios técnicos creados hasta el momento con fines docentes."; este concepto enmarca las potencialidades del medio dentro del proceso de enseñanza, aspecto que se debe tener en cuenta a la hora de su utilización en la docencia.

Sin embargo la eficiencia del uso de la computadora en la docencia no depende solo del número de equipos instalados sino también del tipo de software que se utiliza. Alessi (Francia, Garay y García 2000). plantean que los Entrenadores, Tutoriales, Juegos didácticos, Sistemas expertos para la enseñanza, Simuladores de experimentos y los Evaluadores, son las formas más difundidas de presentarse el software con fines docentes.

Los programas bien diseñados pueden fomentar el aprendizaje conceptual, al ofrecer un contexto que permita a los alumnos extraer los principios generales de los contenidos. Este tipo de enseñanza, "descubrimiento guiado", puede ser muy útil para el aprendizaje de ciertas materias.

I.2 Enseñanza asistida por computadora (EAC)

El proceso de Enseñanza – Aprendizaje se puede ver favorecido por diferentes recursos tecnológicos. En este sentido, conviene recordar que un determinado recurso tecnológico, como cualquier otro medio, presenta la realidad desde una determinada óptica. De ahí la importancia de combinarlo con otros que aporten perspectivas diferentes. Podemos citar como ejemplo el estudio de un entorno concreto, en el que puede utilizarse un programa gestor de base datos para extraer información numérica o textual, y también un documento audiovisual que presente información adicional a través de imágenes.

"Cada medio requiere y desarrolla capacidades específicas. Cada medio privilegia determinadas formas y sistemas de representación y cada medio transmite también una serie de formatos o estructuras que utiliza hábilmente y que le permiten comunicar sus contenidos". (Ramos, 2006)

Por otro lado, en numerosas ocasiones, el acercamiento a la realidad que proporcionan las Nuevas Tecnologías de la Información y las Comunicaciones (TIC) puede ser mayor que el aportado por otros medios: es el caso de la diferencia entre las ilustraciones contenidas en un libro y las imágenes con movimiento y sonido, presentadas por un documento videográfico, sobre un mismo tema.

Hoy en día los maestros que deciden emplear la computación como medio de enseñanza tienen a su disposición una amplia gama de programas que pueden ser empleados por ellos con ese propósito. Unos han sido desarrollados expresamente con ese fin por equipos multidisciplinarios integrados por pedagogos, psicólogos, artistas y programadores, otros por solitarios programadores que se apoyan en sus conocimientos sobre su especialidad para apuntalar su discutible experiencia (o a veces intuición) pedagógica, y otros, son

simples programas comerciales que por algunas de sus características pueden ser empleados con provecho dentro de la actividad docente.

Cada uno de estos programas tiene propósitos específicos, dirigidos a contribuir con el desarrollo de alguno (a veces con más de uno) de los aspectos del proceso docente. Unos pretenden enseñar al alumno un contenido nuevo, otros simulan el desarrollo de un proceso físico, los hay que intentan contribuir al desarrollo de alguna habilidad, intelectual o motora; otros sólo pretenden evaluar los conocimientos del estudiante sobre un determinado contenido.

En dependencia de estas características de los softwares se ha venido estableciendo una agrupación y una clasificación de los mismos tomando como elemento clasificador la función que realizan dentro del proceso docente. Es usual encontrar en la literatura clasificaciones como la siguiente: (Gómez, 2007)

- Tutoriales.
- Entrenadores.
- Repasadores.
- Evaluadores.
- Simuladores.
- Libros electrónicos.
- Juegos Instructivos.
- Multimedia.
- Realidad virtual.

I.3 Las Tecnologías de la Informática y las Comunicaciones (TIC) en la educación.

El impacto social de las TIC toca muy de cerca a escuelas y universidades, propiciando modificaciones en las formas tradicionales de enseñar y aprender. Sin embargo, es perfectamente posible distinguir tres grupos de instituciones escolares: Las que poseen los recursos económicos para adquirir la tecnología y un desarrollo profesional de sus docentes que les permita llevar a cabo una verdadera transformación en la forma de enseñar. Las que aún teniendo la posibilidad de adquirir la tecnología; carece de un claustro preparado para darle un

correcto uso educacional; y finalmente la gran mayoría de instituciones que carecen de recursos económicos para renovar su parque tecnológico al ritmo que impone el desarrollo de este. (Grau, 2007)

Es tarea de los educadores utilizar las TIC como medios para proporcionar la formación general y la preparación para la vida futura de sus estudiantes, contribuyendo al mejoramiento en el sentido más amplio de su calidad de vida. (Días, 2007)

Es necesario que en el ámbito educacional se gane conciencia de que el empleo de estos nuevos medios impondrán marcadas transformaciones en la configuración del proceso pedagógico, con cambios en los roles que han venido desempeñando estudiantes y docentes. Nuevas tareas y responsabilidades esperan a estos, entre otras, los primeros tendrán que estar más preparados para la toma de decisiones y la regulación de su aprendizaje y los segundos para diseñar nuevos entornos de aprendizaje y servir de tutor de los estudiantes al pasarse de un modelo unidireccional de formación donde él es el portador fundamental de los conocimientos, a otro más abierto y flexible en donde la información se encuentra en grandes bases de datos compartidos por todos.

I.4 Modalidades de estudio, Presencial y semipresencial

En la educación superior época actual, al margen de las posibles particularidades que puedan existir en diferentes países, se acostumbra a distinguir, básicamente, entre dos modalidades de estudio; diferentes por el modo de asumir la relación estudiante-profesor: “presencial” y “a distancia”.

La modalidad presencial es entendida generalmente, como aquella donde el proceso de formación tiene lugar a partir de la presencia de los estudiantes y sus profesores, en el mismo lugar, en el mismo tiempo y con altos niveles de carga lectiva semanal, con lo cual se asegura una relación estable y permanente para lograr los objetivos propuestos. Esta modalidad es la más apropiada para estudiantes que dedican todo su tiempo a los estudios y de docentes a dedicación

exclusiva y por tanto, no constituye la respuesta más general al objetivo del pleno acceso que Cuba se ha propuesto alcanzar.

La modalidad semi-presencial viene acompañada de un nuevo modelo pedagógico que a su vez exige el desarrollo de habilidades básicas para acceder a la educación superior y transitar con eficiencia por la universidad. La aplicación de dicho modelo ha venido revelando que algunas de sus cualidades esenciales son similares a las de otros tipos de curso vigentes en los centros del MES, tales como los Cursos para Trabajadores y la Educación a Distancia, esta última con el enfoque que se viene aplicando para el nuevo ingreso desde el curso 2005-2006 y ya generalizado a todo el país en el 2006-2007.

El modelo semi-presencial se ha identificado por el Ministerio de Educación Superior en los últimos cursos académicos como una prioridad en el trabajo metodológico aunque aún no se ha logrado la aplicación de dicho modelo con la profundidad, alcance y calidad que demanda el mismo.

Los medios de enseñanza en la modalidad semi-presencial juegan un importante papel en el proceso de aprendizaje y a través de los mismos debe transitar una parte apreciable de la adquisición de contenidos por parte de los estudiantes.

Entre los medios más utilizados en esta modalidad se encuentran:

Los medios impresos: Textos básicos, guías de estudio que oriente eficazmente en empleo de los textos básicos; guía del profesor; guía de la carrera, guía de asignatura, guía de video, textos complementarios, otros documentos impresos complementarios, etc.

Los medios audiovisuales e informáticos: Videos, transparencias, audio casetes, radio y TV educativa, software educativo, materiales en formato digital, laboratorios virtuales, multimedia, correo electrónico, Internet, plataformas interactivas, etc.

I.5 Plataformas existentes en los entornos educativos

En la esfera educativa han surgido muchas herramientas para ayudar al profesor en su trabajo como las plataformas educativas de formación. Dichas herramientas

son utilizadas tanto en la educación presencial, como a distancia. Esto se logra a través del uso de las nuevas tecnologías en la educación. Ejemplo de estas plataformas son: CLAROLINE, TELEDUC, ILIAS, GANESHA, FLE3, MOODLE, MICROCAMPUS.

Moodle

Es una plataforma de aprendizaje a distancia basada en software libre. Constituye un sistema de gestión de la enseñanza, es decir, una aplicación diseñada para ayudar a los educadores a crear cursos de calidad en línea. En términos de arquitectura, se trata de una aplicación web que puede funcionar en cualquier computadora.

Moodle es lo suficientemente flexible para permitir una amplia gama de modos de enseñanza. Puede ser utilizado para generar contenido de manera básica, avanzada (por ejemplo páginas web) o desde el punto de vista evaluativo y no requiere un enfoque constructivista de enseñanza.

Una de las características más atractivas de Moodle, que también aparece en otros gestores de contenido educativo, es la posibilidad de que los alumnos participen en la creación de glosarios, y en todas las lecciones se generan automáticamente enlaces a las palabras incluidas en estos.

Además las Universidades, podrán poner su Moodle local y así poder crear sus plataformas para cursos específicos en la misma universidad y dando la dirección respecto a Moodle, se moverá en su mismo idioma y podrán abrirse los cursos a los alumnos que se encuentren en cualquier parte del planeta.

Microcampus

Es un gestor de cursos de última generación para el apoyo de la enseñanza presencial y una herramienta esencial para la enseñanza a distancia a través de Internet. Pertenece a una red de centros en Iberoamérica, que desarrollan proyectos educativos basados en la educación a distancia con el uso de las nuevas tecnologías aplicadas a la formación. Permite conocer en detalles el uso por parte de los estudiantes de la plataforma.

CLAROLINE

Es un gestor de educación a distancia que ofrece facilidades de trabajo a estudiantes, profesores y trabajadores de centros educativos, incluye vínculos a sistemas utilizados en el centro donde se usa la plataforma.

I.6 Ingeniería de Software

En la construcción y desarrollo de proyectos se aplican métodos y técnicas para resolver los problemas, la informática aporta herramientas y procedimientos sobre los que se apoya la ingeniería de software con el fin de mejorar la calidad de los productos de software, aumentar la productividad y trabajo de los ingenieros del software, facilitar el control del proceso de desarrollo de software y suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.(Álvarez, 2007)

El objetivo es convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permita obtener un producto final de alta calidad y satisfaga las necesidades y expectativas del cliente.

La Ingeniería de Software (Crawford, 2005) es un proceso intensivo de conocimiento, que abarca la captura de requerimientos, diseño, desarrollo, prueba, implantación y mantenimiento. Generalmente a partir de un complejo esquema de comunicación en el que interactúan usuarios y desarrolladores, el usuario brinda una concepción de la funcionalidad esperada y el desarrollador especifica esta funcionalidad a partir de esta primera concepción mediante aproximaciones sucesivas. Este ambiente de interacción motiva la búsqueda de estrategias robustas para garantizar que los requisitos del usuario serán descubiertos con precisión y que además serán expresados en una forma correcta y sin ambigüedad, que sea verificable, trazable y modificable.

El término ingeniería del software (Herrera, 2007) empezó a usarse a finales de la década de los sesenta, para expresar el área de conocimiento que se estaba

desarrollando en torno a las problemáticas que ofrecía el software. En esa época, el crecimiento espectacular de la demanda de sistemas de computación cada vez más y más complejos, asociado a la inmadurez del propio sector informático (totalmente ligado al electrónico) y a la falta de métodos y recursos, provocó lo que se llamó la crisis del software. Durante esa época muchos proyectos importantes superaban con creces los presupuestos y fechas estimados. La crisis del software finalizó pues se comenzó a progresar en los procesos de diseño y metodologías. Así pues, desde 1985 hasta el presente, han ido apareciendo herramientas, metodologías y tecnologías que se presentaban como la solución definitiva al problema de la planificación, previsión de costes y aseguramiento de la calidad en el desarrollo de software (Herrera, 2007).

Cada año surgen nuevas ideas e iniciativas encaminadas a ello. En combinación con las herramientas, también se han hecho esfuerzos por incorporar los métodos formales al desarrollo de software, argumentando que si se probaba formalmente que los desarrolladores hacían lo que se les requería, la industria del software sería tan predecible como lo son otras ramas de la ingeniería. La dificultad propia de los nuevos sistemas, y su impacto en el negocio, han puesto de manifiesto las ventajas, y en muchos casos la necesidad, de aplicar una metodología formal para llevar a cabo los proyectos de este tipo.

I.7 Metodologías de Desarrollo de Software

Una parte importante de la ingeniería de software es el desarrollo de metodologías y han sido muchos los esfuerzos que se han encaminado al estudio de los métodos y técnicas para lograr una aplicación más eficiente de las metodologías y tener en correspondencia sistemas de mayor calidad con la documentación necesaria en perfecto orden y en el tiempo requerido.

El papel preponderante de las metodologías es sin duda esencial en un proyecto y es el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo.

Atendiendo a las características del proyecto a desarrollar las metodologías de desarrollo de software se dividen en grupos, entre ellos, las llamadas Metodologías Tradicionales y las Metodologías Ágiles.

I.7.1 Metodologías Tradicionales

Al inicio el desarrollo de software era artesanal en su totalidad, la fuerte necesidad de mejorar el proceso y llevar los proyectos a la meta deseada, tuvieron que importarse la concepción y fundamentos de metodologías existentes en otras áreas y adaptarlas al desarrollo de software. Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial que de algo mejoraba la necesidad latente en el campo del software.

Entre las principales metodologías tradicionales se encuentran los ya tan conocidos Rational Unified Process (RUP) y Microsoft Solution Framework (MSF) entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del mismo.

Otra de las características importantes dentro de este enfoque son los altos costos al implementar un cambio y al no ofrecer una buena solución para proyectos donde el entorno es volátil.

Las metodologías tradicionales se focalizan en documentación, planificación y procesos. (Plantillas, técnicas de administración, revisiones, etc.), a continuación se detallan algunas de las más usadas dentro de los métodos tradicionales.(Gacitúa, 2003)

Rational Unified Process (RUP)

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades

de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación. (Jacobson, 2000)

Las cuatro fases del ciclo de vida son:

- Concepción
- Elaboración
- Construcción
- Transición

Las Ventajas más notables son:

- Evaluación en cada fase que permite cambios de objetivos
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases.

Las desventajas fundamentales son:

- La evaluación de riesgos es compleja
- Excesiva flexibilidad para algunos proyectos
- Se pone al cliente en una situación que puede ser muy incómoda para él.
- El cliente deberá ser capaz de describir y entender a un gran nivel de detalle para poder acordar un alcance del proyecto con él.

Microsoft Solution Framework (MSF)

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.

Todo proyecto es separado en cinco principales fases:

- Visión y Alcances.
- Planificación.
- Desarrollo.
- Estabilización.

- Implantación.

La fase de visión y alcances trata uno de los requisitos fundamentales para el éxito del proyecto, la unificación del equipo detrás de una visión común. El equipo debe tener una visión clara de lo que quisiera lograr para el cliente y ser capaz de indicarlo en términos que motivarán a todo el equipo y al cliente. Se definen los líderes y responsables del proyecto, adicionalmente se identifican las metas y objetivos a alcanzar; estas últimas se deben respetar durante la ejecución del proyecto en su totalidad, y se realiza la evaluación inicial de riesgos del proyecto.

La fase de planificación es cuando la mayor parte de la planeación para el proyecto es terminada. El equipo prepara las especificaciones funcionales, realiza el proceso de diseño de la solución, y prepara los planes de trabajo, estimaciones de costos y cronogramas de los diferentes entregables del proyecto.

Durante la fase de desarrollo el equipo realiza la mayor parte de la construcción de los componentes (tanto documentación como código), sin embargo, se puede realizar algún trabajo de desarrollo durante la etapa de estabilización en respuesta a los resultados de las pruebas. La infraestructura también es desarrollada durante esta fase.

En la fase de estabilización se conducen pruebas sobre la solución, las pruebas de esta etapa enfatizan el uso y operación bajo condiciones realistas. El equipo se enfoca en priorizar y resolver errores y preparar la solución para el lanzamiento.

Durante la fase de implantación el equipo implanta la tecnología base y los componentes relacionados, estabiliza la instalación, traspasa el proyecto al personal soporte y operaciones, y obtiene la aprobación final del cliente.

I.7.2 Metodologías Ágiles

Existen varias opiniones tanto a favor como en contra de las metodologías tradicionales de lo que se genera un nuevo enfoque denominado, métodos ágiles, que nace como respuesta a los problemas detallados anteriormente y se basa en dos aspectos puntuales, retrasar las decisiones y la planificación adaptativa; permitiendo potencial aún más el desarrollo de software a gran escala.

Como resultado de esta nueva teoría se crea el Manifiesto Ágil cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

Entre los principales métodos ágiles están XP (eXtreme Programming), Scrum, Iconix, Cristal Methods, Dynamic Software Development Method (DSDM), Feature-driven Development, Lean Software Development.

Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Nos lo proponen porque para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significar reducir su coste.

Retrasar las decisiones y Planificación Adaptativa es el eje en cual gira la metodología ágil, el retrasar las decisiones tan como sea posible de manera responsable será ventajoso tanto para el cliente como para la empresa, lo cual permite siempre mantener una satisfacción en el cliente y por ende el éxito del producto, las principales ventajas de retrasar las decisiones son:

- Reduce el número de decisiones de alta inversión que se toman.
- Reduce el número de cambios necesario en el proyecto.
- Reduce el costo del cambio.

La planificación adaptativa permite estar preparados para el cambio ya que se ha introducido en el proceso de desarrollo, además hacer una planificación adaptativa consiste en tomar decisiones a lo largo del proyecto, transformando el proyecto en un conjunto de proyectos pequeños.

Esta planificación a corto plazo permitirá tener software disponible para los clientes y además ir aprendiendo del feedback para hacer la planificación más sensible.

EXTREME PROGRAMMING (XP)

Es la más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Scrum

Scrum es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming.

Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto que se está

construyendo a las necesidades del cliente. Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.

Scrum tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. El cliente se entusiasma y se compromete con el proyecto dado que ve crecer el producto iteración a iteración y encuentra las herramientas para alinear el

ICONIX

El proceso ICONIX (Oliva y Patricia, 2007) se define como un proceso de desarrollo de software práctico. Está entre la complejidad de RUP y la simplicidad y pragmatismo de XP, sin eliminar las tareas de análisis y diseño que XP no contempla.

Es un proceso simplificado en comparación con otros procesos más tradicionales, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto. ICONIX presenta claramente las actividades de cada fase y exhibe una secuencia de pasos que deben ser seguidos. Además, está adaptado a patrones y ofrece el soporte UML, dirigido por Casos de Uso y es un proceso iterativo e incremental.

Las tres características fundamentales de ICONIX son:

- Iterativo e incremental: varias interacciones ocurren entre el modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.
- Trazabilidad: cada paso está referenciado por algún requisito. Se define la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos
- Dinámica del UML: la metodología ofrece un uso dinámico del UML como los diagramas del caso de uso, diagramas de secuencia y de colaboración.

Las tareas que se realizan en la metodología ICONIX son:

- Análisis de requisitos
- Análisis y diseño preliminar
- Diseño

- Implementación

Crystal Methodologies

Es un conjunto de metodologías ágiles para equipos de diferentes tamaños y con distintas características de criticidad. Fue propulsada por uno de los padres del Manifiesto Agil, Alistair Cockburn, que consideraba que la metodología debe adaptarse a las personas que componen el equipo utilizando políticas diferentes para equipos diferentes. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores: Crystal Clear (3 a 8 miembros), Crystal Yellow (10 a 20 miembros), Crystal Orange (25 a 50 miembros), Crystal Red (50 a 100 miembros) y Crystal Blue (para más de 100 miembros). Por ejemplo, Crystal Clear, la metodología más ligera de este conjunto, esta dirigida a la comunicación de equipos pequeños que desarrollan software cuya criticidad no es elevada. Tiene asociadas siete características: liberación frecuente de funcionalidad, mejora reflexiva, comunicación osmótica, seguridad personal, atención, fácil acceso para usuario expertos y requisitos para el entorno técnico. (Rodríguez, 2008)

Dynamic Software Development Method (DSDM)

Puede considerarse un marco para el proceso de producción de software, más que una metodología. Nació en 1994 con el objetivo de crear una metodología RAD (Rapid Application Development) unificada. Divide el proyecto en tres fases: pre-proyecto, ciclo de vida del proyecto y post-proyecto especificando de forma rigurosa la arquitectura y gestión del proyecto.

Así, propone cinco fases en el desarrollo del proyecto: estudio de la viabilidad y estudio del negocio que constituyen la etapa de pre-proyecto; y, de forma iterativa, modelado funcional, diseño y construcción y finalmente implementación, además de una adecuada retroalimentación a todas las fases. Sus principales características son interacción con el usuario, poder del equipo de desarrollo, liberaciones frecuentes de funcionalidad, regirse siguiendo las necesidades del negocio, desarrollo iterativo e incremental, adaptación a los cambios reversibles,

fijar el nivel de alcance al comienzo del proyecto, pruebas durante todo el desarrollo y eficiente y efectiva comunicación.

Feature-driven Development

Esta metodología, ideada por Jeff De Luca y Peter Coad, combina el desarrollo dirigido por modelos con el desarrollo ágil. Se centra en el diseño de un modelo inicial, cuyo desarrollo será dividido en función a las características que debe cumplir el software e, iterativamente, se diseñará cada una de estas características. Por tanto, cada iteración consta de dos partes, diseño e implementación de cada característica. Este tipo de metodología está dirigido al desarrollo de aplicaciones con un alto grado de criticidad.

Lean Software Development

Es una metodología de desarrollo dirigida especialmente al desarrollo de sistemas cuyas características varían constantemente. Fue definida por Bob Charette's a partir de su experiencia en proyecto industriales, constituyendo una adaptación para el desarrollo software de las lecciones aprendidas en la industria, en particular, en el sistema de producción automovilista Toyota. La metodología establece que todo cambio en el desarrollo software conlleva riesgos, pero si se manejan adecuadamente pueden convertirse en oportunidades que mejoren la productividad del cliente. Consta de siete principios dirigidos a gestionar los cambios: eliminación de todo aquello que no aporte valor al negocio, conocimiento incremental, toma de decisiones tan tarde como sea posible, deliberar funcionalidad tan pronto como sea posible, poder del equipo, construcción incremental del producto y perspectiva global del proyecto.

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código

Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente Proceso mucho más controlado, con numerosas políticas/normas	Impuestas internamente (por el equipo) Proceso menos controlado, con pocos principios
El cliente interactúa con el equipo de desarrollo mediante reuniones Más artefactos	El cliente es parte del equipo de desarrollo Pocos artefactos
Más roles Grupos grandes y posiblemente distribuidos	Pocos roles Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos Existe un contrato prefijado	Menos énfasis en la arquitectura del software No existe contrato tradicional o al menos es bastante flexible

Tabla 1. : Diferencias entre Metodologías Tradicionales y Ágiles.

El retrasar las decisiones en un proyecto de software permite potenciar el valor del producto tanto para el cliente como al equipo o empresa que desarrolla.

Para que un grupo de desarrollo adopte una metodología ágil debe poseer experiencia trabajando con metodologías tradicionales, ya que la experiencia es la que predomina en los momentos cruciales del proyecto, además debe tener la capacidad de ser equipos auto-gestionados, altamente motivados y con gran innovación

Las metodologías ágiles permiten disminuir costos y brindar flexibilidad a los proyectos de software donde la incertidumbre está presente.

El uso de metodologías tradicionales es esencial al inicio en un equipo de desarrollo de software.

Las metodologías ágiles se deberían aplicar en proyectos donde exista mucha incertidumbre donde el entorno es volátil, donde los requisitos no se conocen con

exactitud, mientras que las metodologías tradicionales obligan al cliente a tomar las decisiones al inicio del proyecto.

I.8 Caracterización de la carrera Ingeniería Informática.

La presente investigación va dirigida a la Carrera de Ingeniería Informática, la cual prepara profesionales integrales comprometidos con la Revolución, cuyo función es desarrollar los procesos relacionados con los sistemas informáticos en las organizaciones, con el propósito de obtener un incremento en la eficacia y la eficiencia de su funcionamiento con técnicas que le permiten analizar el entorno para delimitar los procesos computacionales, la información a procesar y las interrelaciones correspondientes; así como la gestión de proyectos informáticos con un alto nivel de profesionalidad.

Además, está dotado de un conocimiento tecnológico, de organización y dirección de procesos y entidades que le permitan desempeñarse en todos los sectores de la sociedad.

Dicho ingeniero es un profesional de sólida formación tecnológica que se ocupa de la captación, transmisión, almacenamiento, procesamiento, protección y presentación de la información mediante el uso eficiente de computadoras y otros medios. (MES, 2007)

La especialidad de Ingeniero en Sistemas Automatizados de Dirección Técnico Económico (SAD-TE) fue creada en 1976 con el objetivo de formar un especialista que comenzaba a ser necesario a la economía del país y debido a la cantidad de máquinas computadoras electrónicas.

El Plan de Estudio A estaba diseñado de forma tal que este especialista se dedicara a la automatización de los procesos en empresas y dentro de esta se inclinaba hacia los procesos industriales, con el enfoque integral que definían los llamados Sistemas Automatizados de Dirección. La práctica en máquinas computadoras (IRIS 10) era muy poca, las asignaturas relacionadas con SAD-PT eran excesivas y faltaba integración entre ellas, y la formación en análisis y diseño de sistemas resultaba insuficiente. La bibliografía de la especialidad era prácticamente inexistente. El Plan B significó un avance respecto del A

perfilándose mejor los ciclos de asignaturas obteniéndose un mejor completamiento bibliográfico e incluyéndose practicas en máquinas del SUMCE y en microcomputadoras a partir de 1985, lo que permitió ampliar la formación y habilidades de los egresados de la especialidad. Como puede apreciarse, el tiempo dedicado a la formación específica y de la especialidad era solo ligeramente mayor que la mitad del total del tiempo del plan de estudio, mientras que el dedicado a las actividades donde predominaba el trabajo independiente del estudiante era aun bajo.

El Plan C comenzó a aplicarse en el curso 90-91, contando con un tiempo total de 6146 horas, de las cuales se dedicaban a actividades independientes curriculares (prácticas docentes, laborales y trabajo de diploma) 1980 horas (32%). Por otra parte, creció también apreciablemente la cantidad de tiempo que debía dedicar el estudiante al trabajo con computadoras personales en la ejecución de los proyectos y trabajos extraclases.

Luego de seis cursos de aplicación del Plan C se han obtenido resultados satisfactorios en la calidad de la preparación de este graduado, pero a la vez se constata la necesidad de introducir nuevas modificaciones en dicho plan de estudio que lo adapten a los nuevos requerimientos. La nueva propuesta de plan de estudio D contribuye a la formación de los recursos humanos profesionales que den respuesta en el siglo XXI a las necesidades planteadas. Los pasos seguidos en las sucesivas modificaciones de este, sumado a aspectos como el cambio en los niveles de abstracción al manejar recursos y herramientas informáticas, la diversidad de enfoques locales en cuanto a tecnologías, lenguajes de programación y áreas de aplicación, el surgimiento de la carrera en todas las provincias, así como enfoques de esencialidad, presencialidad y flexibilidad exigidos. (Mes, 2007)

I.9 Determinación de necesidades del estudiante de la carrera Ingeniería Informática para el análisis y diseño de software.

Premisas para el diagnóstico.

En esta etapa se precisaron como premisas:

- Incorporar a los participantes facilitando un proceso de autorreconocimiento en su realidad.
- Partir de las prácticas cotidianas y su análisis crítico.
- Evaluación del comportamiento de los estudiantes en los proceso de análisis y diseño de software a partir de la observación

Diagnóstico

Se observaron 10 actividades entre discusión de proyectos de curso de las asignaturas Ingeniería de software I, II y discusión de los resultados de las prácticas profesionales IV.

A continuación se exponen las deficiencias señaladas en estas actividades.

- Carecen de calidad las etapas de análisis y diseño de software, arrojado en la falta de eficiencia de los productos finales.
- La generalidad de los proyectos son desarrollados utilizando la metodología RUP.
- En el proceso de desarrollo de software aplican ingeniería inversa, es decir, implementan y luego realizan la documentación.

Al revisar los documentos normativos de la carrera ingeniería informática se detectaron las siguientes regularidades:

- En los programas de la disciplina Ingeniería y Gestión de software (Anexo 4), no se obliga al uso de una metodología específica, sin embargo los profesores solo utilizan la metodología RUP para ejemplificar.
- El modelo del profesional exige aplicar metodologías modernas para organizar y dirigir el desarrollo, implantación y puesta en marcha de los sistemas informáticos.

Se aplicó entrevista a los estudiantes en relación con la preparación que reciben en la carrera en cuanto a la aplicación de diversas metodologías de desarrollo de software, las cuales evidencian que la preparación de estos en este sentido es elemental.

Los resultados permitieron valorar el estado actual de la preparación de los estudiantes en temas relacionados con metodologías de desarrollo de software ágil. Se pudo conocer que:

De 30 estudiantes entrevistados.

- 92% consideran que fue insuficiente el tratamiento de los temas relacionados con la selección de metodologías a utilizar.
- 89% consideran que es necesaria para el análisis y diseño de software.
- 90 % consideran que el tema no es jerarquizado por la carrera.
- 90 % plantean que no saben trabajar con metodologías ágiles.

De 5 docentes de la disciplina Ingeniería y Gestión de Software entrevistados.

- 100% consideran que es necesario el tratamiento de las metodologías de desarrollo de software ágil a partir de las dificultades de los estudiantes en la práctica.
- 21 % consideran que no se jerarquiza ni se entrena en las metodologías de desarrollo de software ágil.

Por las razones antes expuestas se decide implementar el curso con los contenidos referentes a metodologías ágiles y específicamente con la metodología SCRUM, por adaptarse a la mayoría de los proyectos de software actuales. Lo que implica insertar los contenidos de metodologías de desarrollo ágil en el currículo optativo pues estas satisfacen necesidades de aprendizaje, se obtiene una adecuada capacidad de desempeño y de aprender más, de responder a nuevas oportunidades, de adaptarse al cambio y al desarrollo y de participar de forma óptima en el desarrollo de software. Esto responde a las exigencias por parte de la disciplina Ingeniería y Gestión de Software que entre sus objetivos precisa:

- Desarrollar en los estudiantes la necesidad de búsqueda de información adicional acerca de los avances en las técnicas e instrumentos para el desarrollo de un software y en la capacidad de asimilar y desarrollar nuevas tecnologías, contribuyendo de esta forma a la creación de un fuerte espíritu de autopreparación.
- Aplicar metodologías modernas para organizar y dirigir el desarrollo, implantación y puesta en marcha de los sistemas informáticos.(MES, 2007)

Del análisis de todo lo anterior la autora de esta investigación pudo asumir la toma de decisiones desde aspectos claves expresados en el diseño de un módulo instruccional para la elaboración de la propuesta de curso de metodologías ágiles con SCRUM montada en la plataforma de teleformación Moodle.

I.10 Módulo Instruccional

¿Qué es un módulo instruccional?

Un módulo instruccional es un material didáctico que contiene todos los elementos que son necesarios para el aprendizaje de conceptos y destrezas al ritmo del estudiante y sin el elemento presencial continuo del profesor. (Vergel, Brenda, 1998)

I.10.1 El Diseño Instruccional

El diseño instruccional, es un proceso fundamentado en teorías de disciplinas académicas, especialmente en las disciplinas relativas al aprendizaje humano, que tiene el efecto de maximizar la comprensión, uso y aplicación de la información, a través de estructuras sistemáticas, metodológicas y pedagógicas. Una vez diseñada la instrucción, deberá probarse, evaluarse y revisarse, atendiéndose de forma efectiva las necesidades particulares del individuo.

El Diseño Instruccional es una metodología de planificación pedagógica, que sirve de referencia para producir una variedad de materiales educativos, adecuados a las necesidades estudiantiles, asegurándose así la calidad del proceso de aprendizaje.

El diseñador instruccional tiene que entender las debilidades y fortalezas de cada teoría de aprendizaje para poder optimizar su uso en el diseño de la estrategia adecuada.

Los fundamentos de las teorías de aprendizaje son:

Conductismo: Se basa en los cambios observables en la conducta del sujeto. Se enfoca hacia la repetición de patrones de conducta hasta que estos se realizan de manera automática.

Cognoscitivismo: Se basa en los procesos que tienen lugar atrás de los cambios de conducta. Estos cambios son observados para usarse como indicaciones para entender lo que está pasando en la mente del que aprende.

Constructivismo: Se sustenta en la premisa de que cada persona construye su propia perspectiva del mundo que lo rodea a través de sus propias experiencias y esquemas mentales desarrollados. El constructivismo se enfoca en la preparación del que aprende para resolver problemas en condiciones ambiguas.

I.10.2 Fases del Diseño Instruccional

Las fases del Diseño Instruccional se resumen en la siguiente figura.

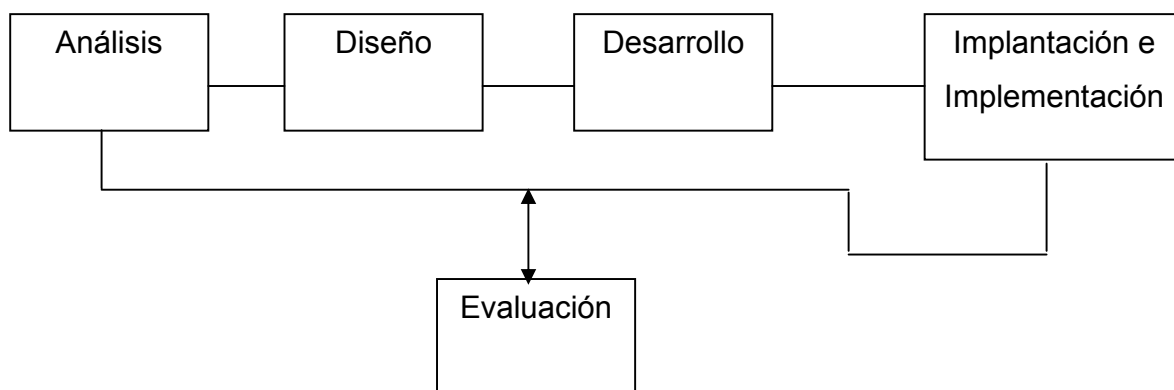


Figura 1. Fases del Diseño Instruccional según el Modelo ADDIE. Fuente:
(González, 2006)

El **Análisis** constituye la base para las demás fases del diseño. En esta fase se define el problema, se identifica la fuente del problema y se determinan las posibles soluciones.

El **Diseño** se utiliza el producto de la fase de análisis para planificar una estrategia y así producir la instrucción. En esta fase se realiza un bosquejo de cómo alcanzar las metas instruccionales. Algunos elementos de esta fase incluyen hacer una descripción de la población a impactarse, llevar a cabo un análisis instruccional, redactar objetivos, redactar ítems para prueba, determinar cómo se divulgará la instrucción y diseñar la secuencia de la instrucción.

En la fase de **Desarrollo** se elaboran los planes de la lección y los materiales que se van a utilizar. En esta fase se elabora la instrucción, los medios que se utilizarán en la instrucción y cualquier otro material necesario, tal como guías de una lección.

En la **Implantación e Implementación** se divulga eficiente y efectivamente la instrucción. En esta fase se propicia la comprensión del material, el dominio de destrezas y objetivos, y la transferencia de conocimiento del ambiente instruccional al ambiente de trabajo.

En la **Evaluación** se evalúa la efectividad y eficiencia de la instrucción. La Evaluación está presente en todas las fases del proceso instruccional.

I.10.3 Descripción del Módulo instruccional. Fundamentación

El modelo sigue un concepto pedagógico, basado en las diferentes teorías y modelos educativos que han tenido vigencia a través del tiempo. Por otra parte, procura la recolección de los contenidos por parte del docente de una manera simple mediante el uso de una serie de tablas que facilitan el trabajo de diseño instruccional del profesor. (Díaz, Ramírez y Talía, 2002)

A continuación se relacionan las secciones de que se compone el modelo las cuales se describen posteriormente:

- 1- Estructura del Curso: El primer paso consiste en determinar la organización global del curso, de la cual depende la secuencia lógica y funcional de los diferentes elementos que la conforman, entre los que se encuentran los materiales de enseñanza.
- 2- Información general del curso: Esta sección está constituida por los datos generales del curso; tales como la ubicación curricular, la introducción, objetivos generales, fundamentación, a quién va dirigido, contenido, temario, dinámica de trabajo, sistema de evaluación plan del curso, prácticas y actividades, bibliografía y glosario. La correcta integración y aclaración de toda esta información es de gran importancia, para el desarrollo del Curso ya que de ella dependerá la adecuada ubicación del estudiante.

- **Ubicación curricular del curso:** Es la especificación de los datos del curso, semestre al que pertenece el curso, tipos de cursos relacionados con éste y duración. Todos estos datos permiten al estudiante conocer las características del que está por comenzar. Esta información es además importante dado que es la manera de ubicar al estudiante en su carrera y la relación con las demás materias que ha tomado, tomará después, o está tomando en paralelo con la materia en cuestión.
- **Introducción del curso:** En esta parte se realiza la presentación del panorama general del curso y los temas de estudio que abordarán durante el desarrollo del mismo, esto se realiza con la finalidad de que el estudiante inicie el curso con información suficiente, como para saber a que se enfrentará en este.
- **Objetivos:** Es la determinación y presentación de los objetivos del curso, la exposición ordenada de estos permite al estudiar saber cuales son las habilidades, actitudes y conocimientos que desarrolle a lo largo del curso.
- **Fundamentación:** Es una de las partes esenciales del curso, ya que presenta al estudiante la razón por la cual debe tomar el curso. Esto es parte del sistema motivacional. Un estudiante mostrará una mayor disposición al estudio y al aprendizaje si le resulta claro de qué le servirá revisar esa unidad de estudio ó curso, la fundamentación es importante porque da un sentido al proceso de aprendizaje del alumno.
- **A quién va dirigido:** Es la caracterización del estudiante que ingresará al curso, menciona las principales habilidades, actitudes, recursos y conocimientos que debe poseer dicho estudiante para que su desempeño sea el más satisfactorio en el desarrollo del curso. De una adecuada selección de las personas que pueden tomar un curso dependerá el éxito del mismo.

- **Contenido:** Es la presentación concreta del tema principal del curso, a través de la cual se pretende que el alumno aborde de manera general, la problemática que se le presentará a lo largo del curso. Tiene la intención de despertar el gusto por la investigación dentro y fuera de los materiales que se han seleccionado en el curso, con la finalidad de que el estudiante busque sus propias respuestas desarrollando y fortaleciendo su capacidad de auto- aprendizaje. Esta presentación es acompañada por un esquema cognoscitivo que permita al estudiante partir de los contenidos generales a los particulares, logrando así un conocimiento claro de los componentes temáticos del curso.
- **Temario:** Es la presentación ordenada de las unidades que constituyen el curso, lo que permitirá al estudiante conocer los contenidos temáticos que se abordarán a lo largo del curso.
- **Dinámica:** Es la explicación detallada de la secuencia en la cual se realizan las actividades que se incluyen en el curso.
- **Sistema de Evaluación:** Es la especificación de los criterios por los cuales será evaluado el desempeño del estudiante, debe presentarse también los cronogramas de las posibles fechas de las sesiones de evaluación o condiciones que deban cubrirse para tener derecho a presentar una evaluación.
- **Plan del curso:** Es la organización y descripción exhaustiva de cada una de las actividades principales del curso.
- **Bibliografías:** Es la presentación del listado de los materiales bibliográficos, básicos y complementario.
- **Refuerzo del aprendizaje:** Es la presentación del resumen general de los materiales vistos en todo el curso, su función es reforzar y retroalimentar los conocimientos adquiridos por el estudiante durante el curso.
- **Glosario:** Es el listado en orden alfabético de las palabras poco comprensibles o técnicas, acompañadas de sus significados.

Conclusiones

En este capítulo se abordaron aspectos teóricos del tema que se va a analizar, exponiendo los principales conceptos asociados al dominio del mismo. Se describe el contexto donde se enmarca, las características y dificultades que lo acompañan, así como específicamente lo relacionado con la diversidad de metodologías de desarrollo de software existentes.

Capítulo 2: Fundamentación y descripción de la solución propuesta.

En este capítulo se describen los aspectos relacionados con el diseño del curso, explicando la necesidad del mismo y detallando las secciones que componen el módulo instruccional.

Además se presentan los fundamentos teórico – metodológicos que interesan para la elaboración de la propuesta de curso de metodologías ágiles con Scrum para la carrera de ingeniería informática.

2.1 Fundamentación de la solución propuesta.

En este apartado se precisan las concepciones del autor acerca del proceso para conformar la lógica científica que sustenta la elaboración y diseño del modelo instruccional del curso como tipo específico de intervención teórica y práctica orientada a la preparación de los estudiantes.

Desde el punto de vista metodológico, el curso presupone el uso de diferentes métodos y formas: la problematización de la práctica, la modelación, simulación de casos reales y el trabajo en equipo. Por su naturaleza, no sólo se incluyen los contenidos de orden técnico, sino se explicitan las concepciones acerca de las formas, métodos y medios que podrán ser utilizados en el desarrollo de los temas que se seleccionen. Los contenidos de este curso emergen de la dinámica entre necesidad, potencialidad; lo normativo y las exigencias de la práctica. Esto explica que se parta de la determinación de sus necesidades y potencialidades tomando como referentes las habilidades en el análisis y diseño de software que debe desarrollar un futuro ingeniero informático.

Asimismo, la evaluación general del modelo que incluye las acciones formativas constituye un aspecto clave para retroalimentar a los implicados acerca de la marcha del proceso, por tanto las dimensiones e indicadores a utilizar para concretar esta exigencia responderá a la lógica del proceso y a la explicación e interpretación de los resultados expresados en cambios, transformaciones de acuerdo a la constante evolución de la industria de software.

En cualquier caso el modelo deberá sustentarse en los siguientes fundamentos:

Filosófico

La conceptualización que se realiza del curso parte de la consideración de que la dialéctica materialista como teoría que orienta el desarrollo y lógica del conocimiento es el fundamento que sustenta el proceso de formación del ingeniero informático. Es a partir de ella que puede conseguirse una visión y práctica integradora del proceso de formación con una orientación acertada. De este análisis se advierte la concepción dialéctico materialista de la actividad y la comunicación, como condición inherente al ser humano, quien de modo consciente actúa sobre él y su entorno e interactúa con los restantes contemporáneos.

Sociológico

La formación del estudiante se asume a partir del modelo de profesional del Ingeniero Informático que se aspira, por lo que se infiere la necesidad de colocarlo en condiciones óptimas de acuerdo con las exigencias de su desempeño profesional. Se concreta en la relación actividad comunicación en la atención a la relación sujeto /contexto, así como las vías y formas mediante las cuales influye en el desarrollo social.

Psicológico

Desde el enfoque histórico cultural que permite asumir el papel de la educación en la conducción del desarrollo, parte de la consideración de que el aprendizaje es un proceso social vinculado al desarrollo del sujeto y se pone de manifiesto en las relaciones entre pensamiento y actividad, aprendizaje y desarrollo, relación de lo individual y lo social, y tiene como referentes, prestar atención a la diversidad individual y grupal en el proceso de apropiación del conocimiento y en la evaluación del resultado, incentivar el estímulo al aprendizaje permanente como elemento básico para el desempeño profesional; la concepción del desarrollo desde la articulación entre exigencias socioeducativas, contexto de actuación y las necesidades de los sujetos.

Didácticos y Pedagógicos

Se parte del conocimiento científico de la realidad para comprenderla y transformarla, incluye por tanto la toma de decisiones acerca de objetivos contenidos, formas y métodos pero su componente más dinámico está en la articulación de estos componentes con el perfil del estudiante, tareas de su desempeño y contextos de actuación, de manera que le permita asumir sus funciones profesionales de forma creadora, problematizadora y transformadora.

Epistemológico

Se asume la formación del estudiante desde los contenidos específicos de la ingeniería y gestión de software. Construcción que se asume en su integridad e interrelación con los contenidos necesarios para el desempeño de este. Esta condición supone el saber, estrechamente vinculado a la práctica, como una fuerza social transformadora que promueve nuevas formas para el perfeccionamiento del desempeño del estudiante universitario a través de un modelo instruccional. Se concibe como un sistema de conocimientos en desarrollo a través de la aplicación sistemática de métodos.

2.2 Pasos en la elaboración de la propuesta:

- 1. Definición de las necesidades:** para realizar la propuesta de curso de metodologías ágiles con SCRUM en la carrera ingeniería informática, se debe partir de su caracterización inicial lo cual incluye al profesor, al estudiante y Modelo del Profesional y a partir de aquí determinar las fortalezas, debilidades, oportunidades y amenazas que tiene el proceso de enseñanza aprendizaje para implementar el curso, en correspondencia se aplican determinados instrumentos (observación, entrevistas, encuestas, entre otros) para determinar las necesidades de la carrera, el sistema de contenidos y definir el accionar metodológico.
- 2. Diseño de la propuesta de curso:** está estructurada de acuerdo a las secciones del modelo instruccional (Estructura del Curso, Información general del Curso, Ubicación curricular del Curso, Introducción del Curso, Objetivos,

Fundamentación, A quién va dirigido, Contenido, Temario, Dinámica, Sistema de Evaluación, Plan del Curso y Bibliografías).

3. **Aplicación de la propuesta:** el curso se incluye en el currículo optativo en la modalidad de estudio semipresencial, montado en la plataforma de formación Moodle con un módulo instruccional.
4. **Valoración de la propuesta:** en este paso es importante constatar los resultados que se van obteniendo con la influencia de las tareas planificadas para el desarrollo del curso. Concluido el proceso de ejecución de las tareas es importante recoger la reflexión y autoevaluación que hacen los docentes y estudiantes para poder conocer si en realidad los estilos de enseñanza que se están empleando permiten al alumno la transferencia de conocimientos, comparar, reflexionar, arribar a conclusiones, investigar e integrar conocimientos y aplicarlos a su especialidad como consecuencias de los cambios en la dirección del aprendizaje.

2.3 Propuesta del Módulo Instruccional

2.3.1 Estructura del Curso

El curso está estructurado en dos módulos.

Módulo Teórico	Módulo Práctico
Tema 1: Metodologías de desarrollo Ágil.	Orientación y desarrollo del proyecto
Tema 2: Introducción a la metodología Scrum.	

Tabla 2. Estructura del Curso.

2.3.2 Información General del Curso

Ubicación Curricular

El curso se ubicará en la disciplina Ingeniería y Gestión de Software para la variante D del plan de estudios, dentro del currículum optativo, el cual está compuesto por los siguientes cursos:

CURRÍCULO OPTATIVO				
Cursos	Horas	Evaluación	Año	Semestre
Gestores de base de datos propietarios y no propietarios	42			
Lenguajes de Programación propietarios y no propietarios	42			
Patrones de diseño y de arquitectura	42			
Herramientas de modelación	42			
Calidad de Software	42			
Validación y Verificación	42			
Gestión de Proyectos	42			
Control de Configuración y Cambio	42			
Compiladores	42			
Programación Paralela	42			
Curso Metodología Ágil con Scrum	44	PC	4to	2do

Tabla 3. Ubicación Curricular.

Introducción del Curso

El curso “Metodologías Ágiles con SCRUM” surge a raíz del creciente desarrollo científico tecnológico que ha provocado el abaratamiento del hardware y a la existencia de entornos integrados de desarrollo baratos y de rápido manejo, que ha llevado a que cada vez la necesidad de desarrollo de productos de softwares es mayor y en el menor tiempo posible. La facultad de informática de la universidad de Cienfuegos desde sus inicio ha impartido las asignaturas de

Ingeniería de Software I, II y III apoyados en la Metodología de desarrollo de software Rational Unified Process (RUP) por las características de esta metodología que hace que sea una buena opción, es este caso, con fines docentes.

A modo de introducción se presentará el panorama general del curso donde se abordarán los temas de estudio durante el desarrollo del mismo, se explicará el sistema de evaluación y cómo se desarrollará el proyecto, esto se realizará con la finalidad de que el estudiante inicie el curso con información suficiente como para saber a qué se enfrentará en este.

Objetivos

Objetivo General:

Fundamentar los presupuestos teóricos, científicos y metodológicos de las metodologías ágiles específicamente SCRUM, para que los estudiantes sean capaces de aplicarlas en el desarrollo de productos informáticos.

Objetivos específicos

1. Apropiarse del marco conceptual de las metodologías ágiles y de SCRUM en particular.
2. Contribuir al desarrollo del pensamiento dialéctico de los estudiantes mediante la solución durante el análisis de los sistemas informáticos.
3. Contribuir al desarrollo del trabajo independiente y creativo de los estudiantes mediante la solución de los problemas que se presentan en el análisis de los sistemas informáticos.
4. Contribuir al desarrollo del espíritu de autopreparación mediante la exigencia del uso de bibliografías especializadas en técnicas modernas de análisis.
5. Contribuir a desarrollar en los estudiantes hábitos de observación de la calidad estética de los resultados de su trabajo.
6. Desarrollar en los estudiantes hábitos de trabajo en equipo que les permitan acometer tareas con esas características.

7. Estudiar el sistema organizativo e informativo de una entidad productiva, de servicio o técnica a cualquier nivel, para poder determinar si es factible el logro de mejoras mediante la transformación de sus procesos y la aplicación de un producto de software, o una combinación de herramientas propias de las tecnologías de la información.
8. Utilizar instrumentos CASE y técnicas de prototipo en el proceso de producción de un producto de software.
9. Estudiar una metodología moderna de análisis y diseño de software y aplicarla en la ejecución de las etapas tempranas de un proyecto informático.

Estos objetivos se implementan a través de varios principios de diseño que se explican en detalles a continuación.

Principio de diseño

Como principio general de diseño del curso, se buscó que cada actividad a los que los alumnos se expusieran expresara algún valor, principio o práctica de SCRUM, buscando aplicarlo de la manera más completa y cercana al “libro”, en este caso, “Scrum desde las Trincheras”, lo que constituye una aplicación directa del principio constructivista “Aprender Haciendo”. A continuación se describe cómo esto es posible.

Aplicación de valores

Los principios de agilidad, retroalimentación y responsabilidad asumida son aplicados prontamente: luego de una breve introducción teórica sobre las bases de las metodologías ágiles, los alumnos se involucrarán rápidamente en el descubrimiento y comprensión de la metodología al tener que ellos mismos preparar y luego dictar charlas sobre diversos aspectos de SCRUM para sus compañeros, las que en el momento de ser realizadas son mejoradas a partir de las consultas y del aporte de los alumnos y del profesor.

El principio de “adaptación local” se refuerza a través de pedir a los alumnos ensayos en los que van evaluando la aplicación y utilidad de las prácticas de SCRUM, para finalmente proyectar su uso en la vida profesional

Principio de dedicación de Tiempo fijo

El curso tiene definida una carga estimada de 20 horas de trabajo por semana para el módulo 2, con 4 horas de sesiones presenciales, destinándose 16 horas para el desarrollo en equipo y quedando el trabajo en casa destinado principalmente a la investigación y preparación de del proyecto final.

Al contar con estos tiempos, se crea un panorama real del desarrollo de software. Por ejemplo, si en la vida real un sprint SCRUM dura de 2 a 3 semanas de trabajo de un equipo en jornada de 4 horas, y un entregable se genera en 3 sprint, con el objetivo de que los estudiantes puedan experimentar en la práctica un desarrollo iterativo. Asimismo, en el curso se motiva a los alumnos a entregar software útil para el cliente al fin de cada ciclo, homologando así los tiempos de una iteración con los de la generación de un entregable. Si los alumnos no logran un producto al final del ciclo, el profesor mediará la relación entre ellos y el cliente para que el resultado final del proyecto sea un software realmente funcional y valioso.

Planificación del desarrollo

Para el segundo módulo del curso los alumnos y clientes (dueño del producto) participan de un taller explicativo. En las clases posteriores, y siguiendo lo aprendido en el taller realizan la pila del producto para la planificación de los sprint, que constituye el producto de partida para su desarrollo.

Organización del equipo

La gestión de equipo es liderada por un SCRUM Manager, que es un alumno escogido por sus propios compañeros.

Autenticidad a través de clientes reales con necesidades reales

Una de las decisiones de diseño del curso fundamentales fue que los alumnos deberían poner en práctica SCRUM haciendo un proyecto para un cliente real, dentro de un contexto real. Esta decisión se justifica por el objetivo principal del taller: poder evaluar en la práctica las metodologías ágiles, para que en definitiva los alumnos puedan desarrollar una opinión propia acerca de lo practicado

Fundamentación Teórica

Las metodologías de desarrollo ágil están adquiriendo gran popularidad en los últimos años. La mayoría de las empresas punteras están comenzando a utilizarlas: Google, Yahoo, Symantec, Microsoft, y una amplia lista más de organizaciones siguen el modelo ágil. Muchos las perciben como la panacea para todos los males de la ingeniería del software. No obstante, la relativa novedad de dichas metodologías hace que aún existan grandes lagunas en el proceso de desarrollo a utilizar siguiendo una filosofía ágil. El presente curso persigue el objetivo de profundizar en la teoría de las metodologías ágiles, específicamente la metodología SCRUM, para que el estudiante logre, al finalizar el mismo, aplicar en la práctica las metodologías ágiles de desarrollo de software.

A quién va dirigido

El curso está dirigido a estudiantes de la carrera de informática 4to año, segundo semestre.

Contenidos y Temarios

Módulo 1		
Tema 1	Introducción a las metodologías de desarrollo ágil. Orientación del debate para el taller.	• Conceptos Básicos. • Objetivo. • Estructura del desarrollo. • Principales Modelos
	Taller Metodologías Ágiles.	Debate sobre diferentes modelos ágiles.
Tema 2	Introducción a la Metodología SCRUM. Orientación del debate para el taller.	• Introducción. • Roles y procesos. • Control y Visibilidad. • Retrospectiva.

		• Planificación, pila del producto y estimación. • Equipo y reuniones. • Sprint y desarrollo de proyectos. • Implementación y escalado de SCRUM. • Herramientas.
	Taller de SCRUM	Debate y ejemplos prácticos.
Módulo 2		
Tema 3	Orientación y desarrollo del proyecto.	Escenarios reales para el desarrollo de proyectos.

Tabla 4. Contenidos y temarios.

Sistema de Evaluación

La evaluación del curso tiene 3 componentes:

- Evaluación de resultado (Ponderación: 30%): Como todo curso de ingeniería de software, no debe omitirse la meta de lograr un software satisfactorio en los tiempos y plazos del proyecto. Se basa en la calidad del producto de software desarrollado, y considera en ella la opinión del propio cliente.
- Evaluación de proceso (Ponderación : 40%): Esta evaluación considera los siguientes aspectos:
 - Cumplimiento de tareas: puntualidad y cumplimiento requisitos estipulados por el profesor
 - Co-evaluación: Evaluación recibida por el alumno de sus pares, de acuerdo a su dedicación, responsabilidad y capacidad de trabajo en equipo. 60
- Evaluación experimental (Ponderación : 30%): en forma grupal e individual los alumnos deberán reflexionar y evaluar la metodología aprendida, considerando:
 - Aplicación: el grado de uso y la dificultad relativa experimentada en cada práctica

Fundamentación y descripción de la solución propuesta

- Utilidad: qué tan útiles fueron las prácticas enseñadas

Las reflexiones y evaluaciones son plasmadas en ensayos de opinión que son entregadas en la mitad y al finalizar el semestre.

Para aprobar el curso, el alumno debe lograr una calificación superior o igual a 3 en los tres componentes especificados. En caso de que el alumno no logre el 3 en alguno de éstos tópicos, deberá realizar un trabajo adicional designado por el profesor.

Plan del Curso

Módulo	Semana	Contenido	Tipo	Lugar	Duración
1	1	Tema 1: Introducción a las metodologías de desarrollo ágil.	C	AE	2
		Taller de Metodologías Ágiles.	CP	AE	2
	2	Tema 2: Introducción a la metodología SCRUM.	C	AE	2
2	3	Tema 3: Orientación y desarrollo del proyecto. Orientación del debate para el taller	CP	Lab	2
	4	Taller Roles y Procesos.	CP	Lab	4
	5	Taller Control y Visibilidad.	CP	Lab	4
	6	Taller Retrospectiva.	CP	Lab	4
	7	Taller Planificación, Pila del producto y Estimación.	CP	Lab	4
	8	Taller Equipo y Reuniones.	CP	Lab	4
	9	Taller Sprint y desarrollo de proyectos.	CP	Lab	4
	10	Taller Implementación y escalado de SCRUM.	CP	Lab	4
	11	Taller Herramientas.	CP	Lab	4
	15	Discusión del Proyecto		AE	4

Tabla 5. Plan del Curso.

Descripción de las actividades.

Las orientaciones que se ofrecen constituyen una guía para la planificación del trabajo del personal encargado de impartir el curso, a modo de propuesta. Este puede ser modificado teniendo en cuenta el cumplimiento de los objetivos propuestos y las características de los estudiantes.

Módulo 1

El tema 1: "Introducción a las metodologías de desarrollo ágil",

Objetivo: Caracterizar las metodologías de desarrollo de software.

El tema tiene como propósito introducir al estudiante en las características de las metodologías ágiles, el por qué de su aplicación, los valores y principios del manifiesto ágil y mostrarle al estudiante algunas prácticas de cómo lograr ser ágil.

Al concluir la introducción se sugiere realizar un taller para analizar diferentes metodologías ágiles existentes aparte de SCRUM, para el cual, se formarán equipos y se le asignará una metodología a cada uno, fomentando el debate.

El tema 2 "Introducción a la metodología SCRUM".

Objetivo: Definir la Metodología SCRUM.

Analizar los conceptos utilizados en la metodología SCRUM.

Ejemplificar las posibles herramientas a utilizar.

Desarrollar habilidades profesionales a partir de la aplicación de metodologías SCRUM.

Pretende brindar los aspectos generales de la metodología, comenzando con una introducción a SCRUM, definiendo aspectos como: roles y procesos, control y visibilidad, retrospectiva en SCRUM, planificación, pila de producto y estimación, equipo y reuniones, sprint y desarrollo de proyectos, implementación y escalado de SCRUM, concluyendo con una breve descripción de las herramientas posibles a utilizar.

Se recomienda realizar un taller para aplicar la teoría impartida en el tema, debido a que este recoge los aspectos fundamentales de la metodología a seguir para aplicar SCRUM a un proyecto de software, esto se debe hacer mediante ejemplos prácticos.

Módulo 2.

Tema 3: Orientación y desarrollo del proyecto.

Objetivo

Aplicar la Metodología SCRUM.

Desarrollar habilidades profesionales investigativas a partir del uso de las metodologías.

Al comenzar este módulo se orientará el proyecto a desarrollar mediante un sistema de talleres que concluirá con la discusión del resultado del proyecto.

Se recomienda desarrollar talleres sobre los contenidos abordados en el tema 2, por la importancia que reviste entrenar al estudiante en el ejercicio de la producción de software, con el objetivo de que el estudiante debata cada uno de los contenidos y lo vayan aplicando a su proyecto final.

Es válido aclarar que el sistema de talleres propuestos para este tema no supone el aprendizaje definitivo en el desarrollo de softwares, sino que constituye una aproximación a la aplicación de metodologías de desarrollo ágil, en especial SCRUM.

Este módulo culmina con la discusión del proyecto donde se evalúan todos los contenidos del curso aplicados a un problema práctico.

Bibliografías

- Flexibilidad con SCRUM, principio de diseño e implementación de campos SCRUM. Juan Palacios.
- SCRUM desde las trincheras. Cómo hacemos SCRUM. Henrick Kniberg.
- Agile –Project-Management-whith-SCRUM. J-Schwaber
- Agiles –Software-Development-SCRUM.
- Agile Estimating and planning. Mike Cohn.
- Desarrollo ágil de software

http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software

- SCRUM Alliance

<http://www.scrumalliance.org/>

- Introducción a Scrum

<http://www.chuidiang.com/ood/metodologia/scrum.php>

- Portal de gestión de proyectos

<http://www.navegapolis.net/>

“Agile Software Development with SCRUM”, Ken Schwaber, Mike Beedle

- “Agile Project Management with SCRUM”, Ken Schwaber

- “Scrum and XP from the trenches”

- “Flexibilidad con Scrum”, Juan Palacio.

http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf

- “User Stories Applied”, Mike Cohn

- “The Zen of Scrum”, Scrum Alliance.

http://www.scrumalliance.org/system/resource/file/75/csm_v5.pdf

- “Peopleware”, Tom DeMarco & Timothy Lister

- www.implementingscrum.com

- agilemanifesto.org

- ScrumAlliance.org

- www.controlchaos.com

- www.agileadvice.com

- www.infoq.com

2.4 Implementacion del curso en Moodle

Para el curso, el profesor se apoyará en la plataforma de aprendizaje el cual le proporcionará una comunicación a distancia con los estudiantes haciendo uso de sus diferentes módulos.

La filosofía planteada de Moodle incluye una aproximación constructiva y contructivista social de la educación, enfatizando en que tanto los estudiantes como los profesores pueden contribuir a la experiencia educativa en muchas formas. (Dougiamas , 2008)

2.4.1 Estructura en módulos de Moodle

Módulo de Tareas

- El profesor especificará la fecha final de entrega de tareas y asignará la calificación obtenida
- Los estudiantes subirán sus tareas (en cualquier formato de archivo) al servidor. Se registrará la fecha en que se ha subido.
- Se permitirá enviar tareas fuera de tiempo, pero el profesor podrá ver el tiempo de retraso.
- Las observaciones del profesor se adjuntarán a la página de la tarea de cada estudiante y se le enviará un mensaje de notificación.
- El profesor tendrá la posibilidad de permitir el reenvío de una tarea tras su calificación (para volver a calificarla)

Módulo de Consulta

Es como una votación. Puede usarse para votar sobre algo o para recibir una respuesta de cada estudiante (por ejemplo, para pedir su consentimiento para algo)

- El profesor podrá ver una tabla que presenta de forma intuitiva la información sobre quien ha elegido qué.
- Se podrá permitir que los estudiantes vean un gráfico actualizado de los resultados
- Módulo Foro
- Hay diferentes tipos de foros disponibles: exclusivos para los profesores, de noticias del curso y abiertos a todos.
- Todos los mensajes deben llevar adjunto la foto del autor.
- Las discusiones podrán verse animadas, por rama, o presentar los mensajes más antiguos o los más nuevos primero.
- El profesor podrá obligar la suscripción de todos a un foro o permitir que cada persona elija a qué foros suscribirse de manera que se le envíe una copia de los mensajes por correo electrónico.

Fundamentación y descripción de la solución propuesta

- El profesor podrá elegir que no se permitan respuestas en un foro (por ejemplo, para crear un foro dedicado a anuncios)
- El profesor podrá mover fácilmente los temas de discusión entre distintos foros.

Módulo Diario

- Los diarios constituyen información privada entre el estudiante y el profesor.
- Cada entrada en el diario podrá estar motivada por una pregunta abierta.
- La clase entera podrá ser evaluada en una página con un único formulario, por cada entrada particular de diario.
- Los comentarios del profesor se adjuntarán a la página de entrada del diario y se enviará por correo la notificación.

Módulo Cuestionario

- El profesor podrá definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios.
- Las preguntas podrá ser almacenadas en categorías de fácil acceso, y estas categorías podrá ser “publicadas” para hacerlas accesibles desde cualquier curso del sitio.
- Los cuestionarios se calificarán automáticamente, y pueden ser recalificados si se modifican las preguntas.
- Los cuestionarios tendrán un límite de tiempo a partir del cual no estarán disponibles.
- El profesor podrá determinar si los cuestionarios pueden ser resueltos varias veces y si se mostrarán o no las respuestas correctas y los comentarios.
- Las preguntas y las respuestas de los cuestionarios podrán ser mezcladas (aleatoriamente) para disminuir las copias entre los alumnos.
- Las preguntas podrán crearse en HTML y con imágenes.
- Las preguntas podrán importarse desde archivos de texto externos.
- Las preguntas podrán tener diferentes métricas y tipos de captura.

Módulo Recurso

- Tendrá la presentación de un importante número de contenido digital, Word, PowerPoint, Flash, video, sonidos, etc.
- Los archivos podrán subirse y manejarse en el servidor, o pueden ser creados sobre la marcha usando formularios Web (de texto o HTML).

Módulo Encuesta

- Se proporcionarán encuestas ya preparadas (COLLES, ATTLS) y contrastadas como instrumentos para el análisis de los temas.
- Se podrán generar informes de las encuestas los cuales incluyen gráficos. Los datos podrán descargarse con formato de hoja de cálculo Excel o como archivo de texto CSV.
- La interfaz de las encuestas impedirán la posibilidad de que sean respondidas sólo parcialmente.
- A cada estudiante se le informará sobre sus resultados comparados con la media de la clase.

2.4.2 Seguridad en Moodle

El profesor podrá añadir una “clave de acceso” para sus cursos, con el fin de impedir el acceso de quienes no sean sus estudiantes. La clave se transmitirá personalmente o a través del correo electrónico personal.

El profesor podrá dar baja a los estudiantes manualmente si lo desea, aunque también existe una forma automática para que permanezcan inactivos durante un determinado período de tiempo (establecido por el administrador). Este debe contar con el control total sobre todas las opciones del curso.

2.4.3 Interfaz de Usuario

En el proceso de enseñanza-aprendizaje, la interfaz es el puente de comunicación entre lo que se quiere enseñar y el estudiante, ya que es la puerta de entrada y la estructura donde se apoya el contenido del curso (Schwier y Misanchuck, 1993). El Moodle tiene una interfaz sencilla, permitiendo sin tener conocimientos previos del uso de la plataforma permite ir avanzando en la construcción del conocimiento,

así como la capacidad de comunicar estabilidad y claridad a través del contenido que éste va descubriendo poco a poco. Ver Anexo 5.

Conclusiones del capítulo.

En éste capítulo, se analizaron los temas relacionados con el diseño instruccional del curso enfatizando en la ubicación curricular, el plan temático, los objetivos del curso, contenidos detallados por cada tema, la dinámica y el sistema de evaluación , así como la bibliografía necesaria para profundizar en el tema.

Capítulo 3: “Validación de la Propuesta”.

En este capítulo se realiza la validación de la propuesta del curso de acuerdo al criterio de especialistas.

Valorar es un ejercicio de rigor en toda investigación, y en este caso se asume como una valoración sustentada en métodos y técnicas científicas que permiten establecer como criterios de confirmación: la pertinencia, factibilidad y credibilidad. Se tomaron en cuenta los criterios de especialistas.

Pertinencia: Oportuna, conveniente y responde a las exigencias actuales.

Factibilidad: Posibilidades de aplicación y las perspectivas que ofrece en el marco de la formación de los estudiantes de la carrera de ingeniería informática.

Credibilidad: Aceptación que acredita la importancia y valor de la propuesta de curso.

Las pruebas no deben limitarse a una validación de carácter técnico (ausencia de errores) sino que también deben realizarse pruebas funcionales con usuarios reales para asegurarse de que la versión cumple los requisitos establecidos y es razonablemente usable (siempre existe una inevitable resistencia al cambio en los usuarios que debe ser tomada en consideración).

3.1 Criterios de validez de la propuesta del curso de Metodologías Ágiles con Scrum.

En interés de la autora la validación se efectuó en dos momentos: primero, con los especialistas para legitimar el valor de los criterios se procedió a registrar en memorias las valoraciones de cada uno y luego se procesó la información determinando regularidades en torno a los criterios dados; y un segundo momento, con los estudiantes que recibirán el curso.

La fiabilidad de los datos se asume porque descansa en la confrontación de la información que se obtiene de los grupos de trabajo, lo cual permitió la reflexión y ajuste del programa. De modo que se incluyeron los resultados de las interpretaciones de la información que se obtuvo en las discusiones y valoraciones grupales e individuales.

Una caracterización de los sujetos participantes en la validación se corresponde con los momentos del proceso y se precisan en la tabla que sigue:

Momento	Características de los grupos.
I. Valoración por los especialistas.	Profesores de la disciplina Ingeniería y Gestión de Software.
II. Valoración de los sujetos implicados.	Estudiantes de 4to año de la carrera de ingeniería informática.

Tabla 6. Momentos del Proceso.

3.2 Selección de los participantes en la valoración.

La objetividad del proceso se consiguió mediante la contrastación de los resultados que se obtienen de la valoración de los especialistas y los sujetos implicados en el proceso de valoración. Por su parte la veracidad de las conclusiones se apoya tanto en la significación de la experiencia de los especialistas que participan y las percepciones que los sujetos implicados poseen del modelo que se propone. Estos constituyen un momento clave de la investigación y se encuentra asociada al modo en que se informa el proceso y en las evidencias que se presentan para apoyar la credibilidad.

Para confeccionar el informe se tuvo en cuenta los criterios en atención de su coherencia, al reflejar, registrar y expresar la variabilidad de los datos. Se flexibilizó la interpretación de los resultados en función de contrastar, complementar o compensar las limitaciones de uno u otro método o técnica de investigación para legitimar las condiciones del proceso.

A continuación, se ofrecen los resultados de cada uno de estos momentos en los que se informa del proceso de valoración. Esta lógica toma en cuenta las condiciones en que se desarrolló la investigación y se explica a partir del esquema siguiente:

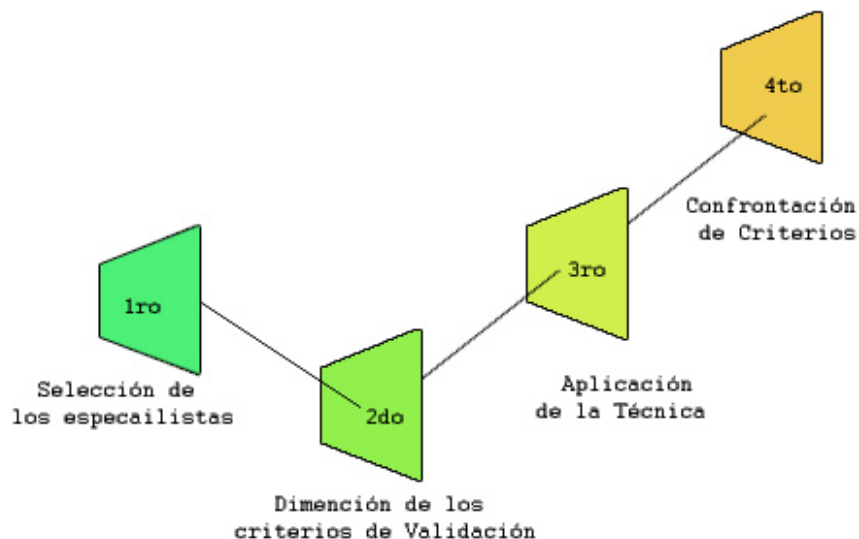


Figura 2. : Secuencia del proceso de validacion de la propuesta.

A continuacion se informa de cada paso y sus reusultados

- **Acerca de los especialistas**

La selección de los especialistas procedió de manera intencional a partir de las referencias de otros investigadores por ser las personas acreditadas y los profesionales de experiencia vinculados con el objeto de esta investigación, el procedimiento se conformó con 5 especialistas de la disciplina ingeniería y gestión de software.

Una vez culminado este paso se procedió a comunicar cómo se desarrollaría el proceso y entregar, el instrumento que debían responder (Anexo 6), al mismo se le incluyó un resumen de la propuesta de modelo instruccional del curso de metodologías ágiles con SCRUM confeccionado.

Los resultados del proceso de valoración por los especialistas fueron los siguientes:

- Acerca de las ideas que sustentan la concepción del modelo y sus componentes. Los resultados de la consulta revelan la aceptación del programa por parte del 100% de los profesores quienes evaluaron de positiva la propuesta. El 20% (1) declaró la propuesta poco adecuada en lo referido a la concepción de uno de sus componentes (Anexo 7).

En relación con las ideas que sustentan la concepción del modelo hubo consenso en todos los aspectos, sin embargo, los resultados que no se ajustan a las categorías de adecuada están relacionados con la opinión de los especialistas al valorar la concepción de los componentes del modelo en cuanto a los contenidos que debían ser incluidos como: otra metodología ágil de desarrollo de software Extreme Programing (XP), que tiene puntos en común con Scrum y que en ocasiones se aplican juntas.

Sugirieron la inclusión de otras bibliografías que han sido material de consulta.

Agregan que esto no solo ayudaría a una mejor orientación, sino también perfecciona la labor investigativa del estudiante.

Con respecto a la evaluación del curso sugieren incluir en la propuesta las oportunidades y posibilidades que se gesten en el propio desarrollo del curso; así por ejemplo se puede incluir la evaluación de un taller donde se analice y diseñe un proyecto utilizando una metodología tradicional y el mismo con una ágil, permitiendo detectar las ventajas y desventajas del uso de las metodologías ágiles.

Consideran que aunque se presentan indicadores para evaluar su resultado, sería provechoso pensar en una manera concreta de comprobar si la preparación recibida por los estudiantes se traduce en el perfeccionamiento de la formación del futuro ingeniero.

- Acerca de la importancia que le conceden los estudiantes, al programa, sus componentes y las posibilidades de aplicación en el marco de su formación profesional.

La valoración por los estudiantes de la carrera de Ingeniería Informática acogió lo relacionado con el curso.

Se empleó el “Grupo de discusión” como técnica de recogida de información. Esta permitió explorar, buscar consenso, generar y evaluar la toma de decisiones didácticas y metodológicas acerca de cómo proceder para ponerlas en práctica. La observación fue empleada en todos los momentos. Ello favoreció el registro de información que describen los procedimientos para su posterior valoración e interpretación.

Para demostrar la validez del programa se tomaron tres criterios a partir de los juicios emitidos por el estudiante. El primero se relacionó con la importancia que le conceden al curso, el segundo con los componentes del curso y el tercero con las posibilidades de aplicación en el marco de su formación profesional.

Durante la presentación del curso se pudo apreciar comprensión pues en el debate en plenaria se recogieron opiniones que se orientan al reconocimiento del significado que tiene esta preparación para ellos.

Asimismo, valoran de gran importancia todos los contenidos que se propone para desarrollarse, pero realzan el valor metodológico de emplear los talleres para llevar a la práctica como recurso didáctico y el intercambio de experiencias prácticas. Refieren como significativo tomar los problemas de la práctica como punto de partida y la viabilidad del montaje del curso en la plataforma Moodle, permitiendo aprovechar las ventajas de la semipresencialidad.

No obstante, existieron múltiples aspectos que generaron preocupación, estos se orientaron a las posibilidades reales de la impartición del curso, la inestabilidad con el servicio del moodle en la Universidad de Cienfuegos. Opinan que sin un compromiso del estudiante respecto a la autopreparación y del docente a asistirlo, no tendría efectividad la propuesta.

De acuerdo con los resultados se apreció que es válida la propuesta en tanto facilita y perfecciona la preparación de los estudiantes de la carrera de ingeniería

informática y se convierte en un instrumento para ampliar las aristas de desempeño del futuro profesional de acuerdo a sus necesidades concretas.

Como resultado de los análisis se manifiesta coincidencia en cuanto a los contenidos, formas, métodos, evaluación y precisan incluir en las modificaciones elementos relacionados con la metodología y organización del curso.

Los resultados advierten las opiniones favorables acerca de la validez del curso para la preparación de los estudiantes de Ingeniería Informática, en sentido general al procesarse la información se concluyó:

- Que resulta oportuno y útil para la toma de decisiones en relación con la proyección de la formación del estudiante.
- Su intencionalidad y organización permite preparar al estudiante para su desempeño profesional.
- Establece un estrecho vínculo entre los acelerados avances científico-técnico y el proceso de enseñanza del futuro ingeniero informático.

Conclusiones del capítulo.

El procesamiento de los criterios evaluativos, su confiabilidad y la utilidad de las valoraciones, reflexiones y recomendaciones que también se derivaron de la revisión de la propuesta por cada uno de los sujetos implicados, demuestran la validez de la misma.

Conclusiones.

- 1- La aplicación de instrumentos para la búsqueda de información, la consulta de investigaciones precedentes sobre el tema constatan que aún no es adecuada la preparación de los estudiantes para la aplicación de metodologías de desarrollo de software ágil a la producción de productos informáticos.
- 2- En la formación del ingeniero informático, el conocimiento profundo de las técnicas y herramientas permiten un adecuado análisis y diseño de software lo que presupone el dominio de las metodologías de desarrollo de software, tanto tradicionales como ágiles, estas con su enfoque centrado en la generación temprana de valor y su acento en el aspecto humano del desarrollo de software facilitan la obtención de productos con mayor prontitud, adaptándose a las necesidades actuales.
- 3- La determinación de las necesidades en el aprendizaje de los estudiantes de la carrera Ingeniería Informática posibilitó el diseño del curso “Metodologías ágiles con Scrum”. lo que permite un adecuado entrenamiento en el dominio y aplicación de las metodologías ágiles para el análisis y diseño de productos informáticos reales.
- 4- La concepción de la propuesta, a partir de los resultados obtenidos en el diagnóstico y las exigencias del programa, constituye, a criterio de los especialistas consultados, una alternativa adecuada para la preparación de los estudiantes en función del perfeccionamiento de la formación del futuro ingeniero informático.

Recomendaciones

A pesar de que los objetivos fueron cumplidos se recomienda:

- Exigir a los estudiantes que se matriculen en el curso “Metodologías ágiles con SCRUM”.
- Socializar el curso a los profesionales informáticos.
- Promover otras investigaciones que permitan al ingeniero informático estar actualizados en los temas relacionados con los avances de la ciencia y la técnica.

Bibliografía

Alvarez, Juana, "Ingeniería de Software," (2007)

<http://www.educando.edu.do/educanblog/index.php?blogId=435> . (julio 2007)

Bartolomé, A. (1995) Nuevas Tecnologías aplicadas a la educación. EDUTEC. La Habana,.—p.295. Bartolomé, A. *Nuevas Tecnologías aplicadas a la educación*. EDUTEC, La Habana.

Beck, Kent y Andres, Cynthia, (2004) "Extreme Programming Explained" Addison-Wesley Professional, 2nd. Edition.

Boehm, B.: (2010). A View of 20th and 21st Century Software Engineering. In: 28th international conference on Software engineering. Shanghai, China

Castejón Costa, J.L. (n.d.). Acercamientos Metodológicos al análisis y representación del conocimiento. Universidad de Alicante. Retrieved from www.psicopedagogia.com.

Castellanos, Doris y otros. (2001). Hacia una concepción del aprendizaje desarrollador. Instituto Superior Pedagógico "Enrique José Varona.

Cao, L., Ramesh, B.: (2008) Agile Requirements Engineering Practices: An Empirical Study. In: IEEE Computer Society Press Los Alamitos, CA, vol 25, pp-60-67

Capiluppi, A., Fernandez-Ramil, J., Higman, J., Sharp, H.C., Smith, N.: (2007) An empirical study of the evolution of an agile-developed software system. In: 29th International Conference on Software Engineering (ICSE), pp. 511{518.

Colectivo de autores. (1998). Documento Normativo para el perfeccionamiento del Sistema Nacional de Educación. La Habana: Editorial Pueblo y Educación.

Colectivo de autores. (1981). Pedagogía. La Habana: Editorial Pueblo y Educación.

Crawford Labrin Broderick, (2005) "Métodos Ágiles: Enfatizando el Tratamiento del Conocimiento Tácito sobre el Explícito,";

www.frcu.utn.edu.ar/deptos/depto_3/34JAIIO/34JAIIO/asse/asse09.pdf.

Declaración de Luis Ignacio Gómez. (2007) <http://www.cubaminrex.cu>

Dimas Hernández y otros. (2006). La nueva Universidad cubana y su contribución a la universalización del conocimiento. ISP Félix Valera.

- Díaz Antón, Maria Gabriela. (2007). *Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática*. <http://www.academia-interactiva.com/ise.pdf>.
- Díaz Camacho, José E.; Ramírez Velásquez, Talía. (2002). Psychology applied to Web course design. San Diego: Alliant Ferré Grau, Xavier. Desarrollo orientado a objetos con UML. Retrieved from <http://www.clikear.com/manuales/uml/introduccion.asp>.
- Dyba, T., Dingsoyr, T.: (2008). Empirical Studies of Agile Software Development: A Systematic Review, Information and Software Technology doi: 10.1016/j.infsof.2008.01.006
- Dougiamas, Martin 2008.
- Espinoza Padierna, Luz Elena. (2005). Tecnología y Educación ¿integración del conocimiento o fragmentación cultural? Paris.
- Eudema. Aparicio, F, & González. (1994). La calidad de enseñanza superior y otros temas Universitarios. ICE de Universidad Politécnica de Madrid.
- Feixas, Mónica. (2001b). Los cambios de la docencia en el profesor universitario. Retrieved from <http://dewey.uab.es/pmarques/dioe/cambios%20docenciaev.pdf#search='Competencias%20docentes%20del%20profesor%20universitario>
- Ferrer López, Miguel Angel. (2005). *La información científico técnica en las transformaciones educacionales*. Maestría en Ciencias de la Educación., Habana.
- Fleeger, P.: 1999 "Software Engineering. 3ed." Prentice Hall,.
- Francia H, Joel. (2006). *Desarrollo de una Aplicación en tres Capas con VS .NET*. <http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art140.asp>
- Gacitúa Bustos, Ricardo A. (2003)., "Métodos de desarrollo de software": El desafío pendiente de la estandarización,"
- Grau Xavier, Ferrá. (2007) *Desarrollo orientado a objetos con UML*. <http://www.clikear.com/manuales/uml/introduccion.asp>.

- Gastón y otros. (1996). Metodología de la investigación pedagógica, La Habana: Editorial Pueblo y Educación.
- Hernández Sampier, Roberto. (2005). Metodología de la Investigación. La Habana: Ed Pueblo y Educación.
- Herrera Urive, Eliécer y Valencia Ayala, Luz Estela, (2007)“Del Manifiesto Ágil sus ValoresyPrincipios,”ScientiaetTechnica,
<http://redalyc.uaemex.mx/redalyc/pdf/849/84934064.pdf>.
- Hurrutiner Silva, Pedro H. (2000). La universidad cubana: El modelo de formación. Félix Valera.
- Ilieva, S., Ivanov, P., Stefanova, E., (2004) “Analyses of an agile methodology implementations”, Proceedings 30th Euromicro Conference, , IEEE Computer Society Press, pags 326 - 333.
- Jacobson, I. (2000) El Proceso Unificado de Desarrollo de software. Addison-Wesley, EE.UU.
- La Educación Superior en el siglo XXI: visión y acción. (1998). UNESCO.
- Labarrere, G Y G. Valdivia. (n.d.). *Pedagogía*. La Habana: Editorial Pueblo y Educación.
- Mann, C., Maurer, F.: (2005). A case Study on the Impact of Scrum on Overtime and Customer Satisfaction. In: Proceedings of the Agile Development Conference (ADC'05). IEEE Computer Society
- Mes, Plan de estudios carrera Ingeniería Informática, Julio, 2007
- Modelo del profesional de la carrera Ingeniería Informática. (2007). MES.
- Nerur, S., Mahapatra, R., Mangalaraj, G.: (2005). Challenges of migrating to agile methodologies. In: Commun. ACM, vol. 48, no. 5, pp. 72-78
- Oliva, San Martin, Carla Rebeca Patricia, (2007) “Uso de la metodología ICONIX”;
<http://www.unsj-cuim.edu.ar/portalzonda/seminario08/archivos/UsodeICONIX.pdf>
- Orjuela Duarte, Ailin y Rojas, Mauricio, (2008) “Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería de Software Educativo”
<http://pisis.unalmed.edu.co/avances/archivos/ediciones/Edicion%20Avances%202008%202/21.pdf>.

- Pérez Rodríguez, Gastón Y Otros. (1996). *Metodología de la investigación social*. La Habana: Pueblo y Educación.
- Ruiz Iglesias, Magali. (1989). *Profesionales competentes: una respuesta educativa*. Instituto Politécnico Nacional.
- Qumer, A., Henderson-Sellers, B.: (2007). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. In: *Information and Software Technology*.
- Ramos, Mirtha. (2007) *Software educativo. Metodología y criterio para su elaboración y evaluación*. [http:// www.uned.ac.cr](http://www.uned.ac.cr)
- Robinson, H. Sharp, L. (2005) "The social side of technical practices in eXtreme Programming and Agile Processes in Software Engineering", *Lecture Notes in Computer Science*. Berlin: Springer Verlag, pages 139-147.
- Richard L. Scheaffer y James T. McClave. (1993) *Probabilidad y Estadística para Ingeniería*. México: Grupo Editorial Iberoamérica.
- SCRUM desde las trincheras.
- Torres Patricio, Letelier.(2007) *Desarrollo de Software Orientado a Objeto usando UML*. <http://www.creangel.com/uml/intro.php>.
- Vecino Alegret, F. (1986). *Algunas tendencias en el desarrollo de la Educación en Cuba*. La Habana: EditorialPueblo y Educación.
- Vigotski., L. S. (1980). *Psicología de la personalidad*. Editorial Progreso.

Anexos

Anexo1

Instrumentos

Entrevista a estudiantes.

Objetivo: Explorar opiniones en estudiantes sobre la aplicación de las metodologías de desarrollo de software en la producción de productos informáticos.

Actividad. Dirigir el intercambio sobre:

- Herramientas teóricas metodológicas que poseen sobre metodologías de desarrollo de software para desarrollar productos informáticos.
- Factores que no favorecen el aprendizaje de las metodologías de desarrollo ágil.

Exigencias metodológicas

Solicitar colaboración con respuestas sinceras.

Crear un clima favorable con adecuada comunicación con el grupo de entrevistados.

Registrar opiniones de manera encubierta.

Anexo 2

Entrevista individual a los docentes.

Objetivo: Explorar el dominio que poseen los docentes sobre los documentos normativos de la carrera Ingeniería Informática así como los resultados de las últimas investigaciones sobre la aplicación de las metodologías de desarrollo de software.

Actividad. Dirigir el intercambio sobre:

- Contenido del programa y orientaciones metodológicas del año en lo referido a la aplicación de metodologías ágiles.
- Posiciones pedagógicas sobre la inclusión de metodologías ágiles de desarrollo de software.
- Necesidades y perspectiva.

Exigencias metodológicas

Solicitar colaboración con respuestas sinceras.

Crear un clima favorable con adecuada comunicación con el grupo de entrevistados.

Registrar opiniones de manera encubierta.

Anexo 3

Modelo del profesional.

OBJETIVOS GENERALES

1. Participar activamente en la vida social demostrando en todas sus acciones una sólida preparación científica, cultural, política y social sustentada en los valores que deben caracterizar las actitudes de un ingeniero informático, asumiendo posiciones patrióticas, políticas, ideológicas, éticas y morales acordes con los principios martianos y marxista leninista en que se fundamenta nuestra sociedad con una conciencia del impacto social y ambiental que se pueden derivar del uso de las tecnologías.
2. Lograr en los egresados una conciencia de productores, contribuyendo además a desarrollar en ellos los hábitos de organización personal y responsabilidad que requieren las actividades relacionadas con su desempeño como profesionales.
3. Consolidar en el egresado un estilo de trabajo que propicie una actuación independiente y creativa para la solución de los problemas que enfrentará, considerando el amplio espectro de equipos multidisciplinarios que integrará y las restricciones que existan en el medio que trabaja.
4. Formar en los egresados una fuerte conciencia económica, especialmente dirigida a considerar adecuadamente la importancia del ahorro en su sentido más amplio, así como desarrollar iniciativas y otras acciones con el fin de contrarrestar los efectos negativos que, en la esfera productiva y de los servicios, ejerce el bloqueo económico, comercial y financiero impuesto a nuestro país.
5. Formar en los egresados una fuerte conciencia jurídica en correspondencia con los marcos legales de desarrollo de la informática, su relevancia como soporte del almacenamiento, gestión y seguridad de la información y el impacto social de estas tecnologías.
6. Propiciar el desarrollo de un estilo profesional de trabajo en el cual sea objeto permanente de atención la calidad de los resultados del mismo, lo que estará

dado fundamentalmente por la eficiencia y eficacia de las soluciones adoptadas y de la documentación técnica generada a tales efectos.

7. Desarrollar en el estudiante un fuerte espíritu de autosuperación que le permita mantenerse actualizado en los avances de la ciencia y la técnica en su campo profesional.
8. Lograr que el profesional se plantee y ejecute su trabajo tomando en cuenta prioritariamente las necesidades e intereses sociales, prestando atención a las orientaciones principales del Partido y Gobierno relacionadas con su perfil ocupacional.
9. Aplicar con creatividad los vínculos con la defensa que se derivan de las características de los contenidos establecidos en los programas de las asignaturas de formación general y del ejercicio de la profesión.

Objetivo de cuarto año

Objetivos educativos:

1. Afianzar la formación de los valores políticos, éticos y morales adquiridos en los niveles precedentes, así como continuar formando y desarrollando los VALORES DE LA PROFESIÓN.
2. Desarrollar el pensamiento y actuación de un profesional como corresponde a un revolucionario comprometido con su sociedad, sobre la base de aplicar los conceptos, leyes, principios y métodos en que se basan las asignaturas del año desde una óptica dialéctica materialista apropiándose del sistema de valores definidos en el perfil del profesional con un alto grado de motivación por su profesión.
3. Desarrollar una formación integral a través de la participación de los estudiantes en las diversas actividades curriculares, extracurriculares, socio - productivas, de extensión universitaria y políticas, organizados por la institución y las demás organizaciones

Objetivos instructivos:

1. Diseñar y poner a punto la arquitectura de un software, efectuar el diseño detallado y la implementación de los componentes o piezas que conforman un software y emplear patrones para la arquitectura, diseño e implementación de un software.
2. Modelar y resolver problemas de toma de decisión aplicando técnicas estadísticas y de la programación matemática, utilizando o desarrollando software para dar solución a problemas específicos.
3. Emplear las técnicas de simulación como herramienta para el análisis del comportamiento de sistemas.
4. Desarrollar los vínculos establecidos con la defensa de las diferentes asignaturas del ejercicio de la profesión.

En este año se realizan dos proyectos de curso asociados a las dos asignaturas principales e integradoras: Ingeniería de Software II (Currículo base) e Ingeniería de Software III (Currículo propio), las que unidas a la práctica laboral de cuarto año, propician que el estudiante adquiera las habilidades mínimas que requiere para su trabajo profesional.

Anexo 4

Programa de la Disciplina Ingeniería y Gestión de Software.

CURRÍCULO BASE				
Asignaturas	Horas	Evaluación	Año	Semestre
Introducción a la Programación	64		1 ^{ro}	1 ^{ro}
Introducción a la Informática	48		1 ^{ro}	1 ^{ro}
Diseño y Programación Orientada a Objetos	80	PC	1 ^{ro}	2 ^{do}
Estructuras de datos	80	E	2 ^{do}	1 ^{ro}
Bases de Datos	80	PC	2 ^{do}	2 ^{do}
Programación Web	56	PC	3 ^{ro}	1 ^{ro}
Ingeniería de Software I	70	PC	3 ^{ro}	2 ^{do}
Ingeniería de Software II	70	PC	4 ^{to}	1 ^{ro}
CURRÍCULO PROPIO				
Bases de Datos Avanzadas	56		3 ^{ro}	1 ^{ro}
Ingeniería de Software III	56	PC	4 ^{to}	2 ^{do}
Introducción a la Gestión de Software	32		1 ^{ro}	2 ^{do}
Total horas	584			
CURRÍCULO OPTATIVO				
Gestores de base de datos propietarios y no propietarios	42			
Lenguajes de Programación propietarios y no propietarios	42			
Patrones de diseño y de arquitectura	42			
Herramientas de modelación	42			
Calidad de Software	42			
Validación y Verificación	42			
Gestión de Proyectos	42			
Control de Configuración y Cambio	42			
Compiladores	42			

Programación Paralela	42			
-----------------------	----	--	--	--

OBJETIVOS EDUCATIVOS

1. Desarrollar en los estudiantes las formas de pensamiento lógico y capacidad de razonamiento, mediante la modelación conceptual y el análisis algorítmico de los problemas, como recursos metodológicos que les posibilite enfrentarse, con un adecuado nivel de profesionalidad, a las exigencias del desarrollo científico-técnico y a los problemas concretos y prácticos que a diario se presentan en nuestro país.
2. Lograr que los estudiantes avancen en la consolidación de una concepción científica del mundo, mediante el estudio de las técnicas de modelación de datos y de programación, analizando el papel del hombre y las máquinas en el desarrollo social, político y económico, en consonancia con los principios del materialismo dialéctico, reconociendo el papel de la práctica social en el proceso del conocimiento
3. Formar en los egresados los hábitos de organización personal y responsabilidad que se requiere en el desarrollo de sistemas automatizados.
4. Contribuir a formar profesionales con una alta calificación en cuanto a las técnicas de desarrollo de sistemas informáticos y que posean como cualidad distintiva la consagración, la sencillez, y la modestia que les permita trabajar en equipo con especialistas de diferentes perfiles y con todo tipo de usuario.
5. Contribuir a que los estudiantes utilicen y desarrollen una forma dialéctica de pensamiento y que lo apliquen consecuentemente en su enfoque sistémico de análisis.
6. Contribuir a crear en el egresado hábitos de disciplina, independencia, creatividad y responsabilidad en las diferentes etapas de desarrollo de un sistema informático.

7. Contribuir a educar a los estudiantes en la consideración de las limitantes que pueden existir en la utilización de los medios técnicos y en la búsqueda de soluciones económicas.
8. Propiciar el desarrollo de un estilo profesional de trabajo en el cual sea objeto permanente de atención la calidad de los resultados de éste, lo que estará dado fundamentalmente por la sencillez y elegancia de las soluciones adoptadas y de la documentación técnica confeccionada.
9. Desarrollar en los estudiantes la necesidad de búsqueda de información adicional acerca de los avances en las técnicas e instrumentos para el desarrollo de un software y en la capacidad de asimilar y desarrollar nuevas tecnologías, contribuyendo de esta forma a la creación de un fuerte espíritu de autopreparación.
10. Lograr que el profesional se plantee y ejecute su trabajo tomando en cuenta prioritariamente las necesidades e intereses sociales.

OBJETIVOS INSTRUCTIVOS

1. Aplicar, con un alto nivel de profesionalidad, los principios de la programación prescriptiva en la automatización de cualquier aplicación.
2. Diseñar las estructuras de datos adecuadas para la solución automatizada de cualquier aplicación.
3. Utilizar mecanismos automatizados de ayuda a la generación de código.
4. Manejar literatura en idioma extranjero con respecto a las técnicas y los productos de programación que utilicen.
5. Analizar un sistema en una organización de base productiva o de servicio o en cualquier otro medio para determinar la factibilidad económica, las tareas a automatizar, los medios técnicos necesarios y la planificación del proyecto.
6. Aplicar metodologías modernas para organizar y dirigir el desarrollo, implantación y puesta en marcha de los sistemas informáticos.

7. Desarrollar los procesos de gestión de software, asegurar la calidad de los productos de software.
8. Dirigir procesos de desarrollo software.
9. Desarrollar en equipo procesos de software.
10. Utilizar herramientas CASE para auxiliarse en todas las etapas de trabajo de desarrollo de sistemas informáticos.
11. Aplicar el enfoque sistemático al desarrollo de los sistemas informáticos.
12. Preparar y entrenar al personal necesario para la puesta en marcha de los sistemas informáticos. Confeccionar la documentación técnica del sistema informático y la orientada a los usuarios finales.

Anexo 5

Ejemplos de Interfaz del curso en Moodle.



1	Tema 1	
	Introducción a las metodologías de desarrollo ágil. Orientación del debate para el taller. Foro de Debate	
2	Tema 2	
	Introducción a la metodología Scrum <i>Dinámica del curso</i> Foro Scrum	
3	Tema 3	
	Introducción y desarrollo del proyecto Orientación del debate para el proyecto Taller, Roles y Procesos Taller, Control y Visibilidad Taller, Retrospectiva Taller, Planificación, Pila del Producto, y Estimación Taller, Equipo y Reuniones Taller, Sprint y Desarrollo de Proyectos Taller, Implementación y Escalado de Scrum Taller, Herramientas <u>Herramientas</u> Sprintometer Scrum for Team plantilla_sprint evaluacion_riesgos_agilidad <i>Dinámica del curso</i> Foro	

Anexo 6

Estimado profesor:

La propuesta de curso de Metodología de desarrollo de software ágil con Scrum para los estudiantes de la carrera de ingeniería informática de la Universidad de Cienfuegos constituye el resultado principal de la tesis que se presenta en opción al título de Master en Nuevas Tecnologías de la Informática y las Comunicaciones para la Educación que se desarrolla en la Universidad Carlos Rafael Rodríguez de Cienfuegos.

Nos sería de gran utilidad que nos exprese su valoración sobre los indicadores que a continuación les exponemos, relacionados con el misma, así como sus sugerencias sobre cada uno de ellos.

1. Los fundamentos teóricos en que se sustenta la propuesta.
2. La concepción general de la propuesta.
3. La estructura de la propuesta.
4. Contenido de los talleres.
5. Ajuste de la propuesta a las necesidades de preparación de los estudiantes.
6. Flexibilidad de la propuesta.
7. Posibilidades de su implementación práctica.
8. Pertinencia de la propuesta.

Después de conocer los indicadores evalúe la propuesta teniendo en cuenta la siguiente escala:

5. Excelente
4. Bien
3. Regular
2. Mal

Anexo 7

Estadístico de los criterios de especialistas.

Indicadores	E	%	B	%	R	%	M	%
Los fundamentos teóricos en que se sustenta la propuesta.	5	100	–	–	–	–	–	–
La concepción general de la propuesta	5	100	–	–	–	–	–	–
La estructura de la propuesta.	4	80	1	20	–	–	–	–
Contenido de las acciones metodológicas.	4	80	1	20	–	–	–	–
Ajuste de la propuesta a las necesidades de preparación de los estudiantes.	5	100	–	–	–	–	–	–
Flexibilidad de la propuesta.	5	100	–	–	–	–	–	–
Posibilidades de su implementación práctica	5	100	–	–	–	–	–	–
Pertinencia de la propuesta	5	100	–	–	–	–	–	–

