

# COMPENDIO DE EJERCICIOS PARA EL DESARROLLO DE HABILIDADES EN LA ASIGNATURA DE INGENIERÍA DE SOFTWARE

## COMPENDIUM OF EXECISES TO DEVELOP OF ABILITIES IN THE SOFTWARE COURSE

*Aneyty Martín García<sup>1</sup>, Dayana Caridad Tejera Hernández<sup>2</sup>, Sahilyn Delgado Pimentel<sup>3</sup>*

1 Universidad de las Ciencias Informáticas, Cuba, amartin@uci.cu

2 Universidad de las Ciencias Informáticas, Cuba, dtejera@uci.cu

3 Universidad de las Ciencias Informáticas, Cuba, sdelgado@uci.cu

**RESUMEN:** *En la carrera de Ingenieros en Ciencias Informáticas específicamente en la asignatura de Ingeniería de Software I es muy importante que los estudiantes desarrollen la habilidad de seleccionar enfoque, modelo(s) y metodología para el proceso de desarrollo de de software. Los profesores que imparten la asignatura han identificado un grupo de deficiencias en desarrollo de esta habilidad en los estudiantes, evidenciándolo en los informes semestrales. Por lo que se decide realizar un compendio de ejercicios para el desarrollo de la habilidad de seleccionar enfoque, modelo(s) y metodología para el proceso de desarrollo de de software en la asignatura Ingeniería de Software 1. Dicho sistema se acoge a las necesidades de autoperparación de los estudiantes.*

**Palabras Clave:** autoperparación, habilidad, compendio de ejercicios

**ABSTRACT:** *In the race of Engineering in Computer Science specifically on the subject of Software Engineering I it is very important that students develop the ability to select focus, model (s) and methodology for the process of software development. The teachers of the subject have identified a group of deficiencias in developing this skill in students evidenciándolo in semiannual reports. So it was decided to perform a compendium of exercises to develop the ability to select focus, model (s) and methodology for the development process of software in the subject Software Engineering 1. This system accommodates the needs of estudiantes.compendio self-preparation exercises to develop skills in software engineering course.*

**KeyWords:** ability, exercises, self-preparation

## 1. INTRODUCCIÓN

El desarrollo científico y tecnológico y el avance de las tecnologías de la informática y las comunicaciones modificaron la sociedad y

acrecentó como nunca antes el consumo de la información digital. A nivel organizacional, estos avances han llevado a la empresa a un contexto donde se produce una gran sobrecarga de información digital, a menudo irrelevante o

sencillamente inoportuna, que va más allá de su ámbito de control y que, lejos de proporcionarle poder, la puede conducir por caminos marcadamente erróneos. El poder entonces, no lo establece la posesión de grandes volúmenes de información digital, sino de aquella de valor, que una vez evaluada y analizada, sea precisa, actualizada y relevante; de aquella que pueda estructurarse, recuperarse y compartirse, en el momento y lugar oportuno y de aquella que pueda introducirse con seguridad en la actividad de la organización para mejorar sus procesos, su desempeño y reducir sus costos[1].

Desafortunadamente, el matrimonio ideal entre información y comunicación no se ha conseguido del todo con el uso de los sistemas automatizados de información, debido a la existencia de una serie significativa de fallos, problemas de calidad y presupuestos sobregirados en el desarrollo de estos. Esta situación ha llevado a comprender que, debido a la complejidad y exigencia de los procesos a automatizar, se requiere de grandes cambios en el proceso de producción de los software y se ha hecho imprescindible la práctica de nuevos métodos, que permitan coordinar, supervisar y establecer el trabajo en equipo entre el grupo de desarrollo de cada proyecto de software y sus clientes, como vía para lograr que los requisitos identificados sean viables, adecuados y permitan decidir, con el mayor por ciento de exactitud posible, qué se desea y qué es posible automatizarse en definitiva[1].

En el proceso de desarrollo de un software existen un grupo de actividades genéricas para todo tipo de sistemas, pero los detalles del proceso de desarrollo de software serán muy diferentes en cada caso, aunque las actividades dentro del marco de trabajo permanecerán iguales. El resultado de estas diferencias conlleva a la existencia de diferentes enfoques de ingeniería de software (IS), diferentes modelo(s) de desarrollo y diferentes metodologías de desarrollo. Por lo que en el ciclo de vida de software es importante la selección adecuada de cada uno de estos elementos para su utilización ya que deben estar en dependencia de las características, contexto del proyecto y la organización que ejecuta el proyecto[2].

Estas prácticas se hacen efectivas en las fases primarias del proceso de desarrollo de un sistema automatizado. En general, cada vez, existe un reconocimiento mayor de su importancia y los riesgos en que se incurre si estas se cumplen de manera incorrecta o insuficiente. Son estas actividades las que definen la dirección y el alcance del proyecto y se realizan según los intereses de la entidad encargada del desarrollo del producto

final[3].

La Universidad de las Ciencias Informáticas[4] no está ajena a este problema ya que la misma se encarga de producir aplicaciones y servicios informáticos, a partir de la vinculación Formación-Producción-Investigación que utiliza los fundamentos de vinculación estudio-trabajo[5]. En el plan de estudios[4] de la carrera se encuentra la asignatura de Ingeniería de Software 1, la misma se imparte en el 3er año de la carrera, la cual tiene como objetivo implementar con eficiencia soluciones informáticas relacionadas con el ciclo de vida de software, acorde al rol que desempeñe en proyectos del área productiva, utilizando las buenas prácticas de Ingeniería de software.

En esta asignatura cada año se ha trabajado en la mejora del diseño y la forma de impartir la misma, pero en un estudio realizado de los informes semestrales de la misma en últimos 5 años, es decir (curso 2010-2011 hasta el curso 2014-2015) se reflejan un grupo de deficiencias que se reflejan a continuación:

- Problemas a la hora de seleccionar el enfoque, el modelo(s) y la metodología a utilizar para el proceso de desarrollo de software.

En las facultades no se logró realizar un cronograma de trabajo basado en la metodología seleccionada, pues no todas trabajaron desde el principio con una metodología en los trabajos de curso y por consiguiente los estudiantes no lograron conocer actividades propias de estas con más profundidad; además se le dedica poco tiempo al desarrollo de dicha habilidad en el plan calendario[6].

Según se plantea en el Programa Analítico[7] los estudiantes deben vencer los siguientes objetivos:

1. Comparar los diferentes enfoques, modelo(s) y metodologías de desarrollo de software a partir de su previa caracterización.
2. Seleccionar una metodología o proceso de desarrollo de software a partir de las características del equipo de desarrollo, las condiciones para el desarrollo del producto y el enfoque de gestión seleccionado.

También se plasma en el Programa Analítico[7] que los estudiantes deben vencer las siguientes habilidades:

1. Comparar enfoques, modelo(s) y metodologías de desarrollo de software.
2. Seleccionar enfoques de gestión del proceso de desarrollo de software.
3. Seleccionar modelo(s) y metodologías de desarrollo de software.

En el Modelo(s) de Planificación y Control del

Proceso Docente 2014-2015 para el desarrollo de estas habilidades y darle cumplimiento a los objetivos sólo se realiza una conferencia (C2), un seminario (S1) y una clase práctica (CP1). Se emplea poco tiempo para el desarrollo de estas habilidades de manera presencial y no se hace alusión más de estos temas hasta la evaluación en el corte del trabajo de curso y luego en la prueba parcial en la semana 9. En el Entorno Virtual de Aprendizaje no existe ningún cuestionario o ejercicio para el desarrollo de la habilidad de seleccionar enfoque, modelo(s) y metodología para el proceso de desarrollo de software.

Por lo que se demuestra que la asignatura no hay suficientes actividades teóricas y prácticas que permitan al estudiante desarrollar la habilidad de seleccionar enfoque, modelo(s) y metodología del proceso de desarrollo de software y actualmente en la asignatura no existen actividades para la autopreparación de los estudiantes en el desarrollo de esta habilidad.

Se propone como objetivo general desarrollar un compendio de ejercicios para contribuir a fortalecer la habilidad de seleccionar el enfoque, modelo(s) y metodología de desarrollo de software, en la asignatura de Ingeniería de Software 1 en la Facultad 1.

## 2. CONTENIDO

### 2.1 Métodos Científicos

Los **métodos teóricos** a emplear para cumplir con las tareas a desarrollar son:

Análisis y Síntesis: Se realizará un análisis para identificar y caracterizar los elementos relacionados con el desarrollo de las habilidades en la carrera en Ciencias Informáticas en la asignatura de ISW1 así como las relaciones entre cada uno de ellos.

Inducción y deducción: Permitirá analizar los datos obtenidos para arribar a generalizaciones sobre la creación de un compendio de ejercicios para el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología de desarrollo de software y la influencia que tiene los mismos en la preparación de los estudiantes.

Sistémico estructural: Se utilizará para el diseño de la propuesta del compendio de ejercicios y la definición de la complejidad de cada una de ellas, orden de consecución y elementos a utilizar para cada caso, de acuerdo a los principios establecidos para el desarrollo de habilidades.

Modelación: Se utilizará para la creación, el diseño y la representación de un compendio de ejercicios para el desarrollo de la habilidad de seleccionar el

enfoque, modelo(s) y metodología de desarrollo de software en la asignatura de Ingeniería de Software 1 en la Facultad 1.

Los **métodos empíricos** a utilizar para obtener información del objeto de estudio son:

Entrevista: Se realizará a los profesores que imparten la asignatura en la UCI y los estudiantes de la Facultad 1 para obtener la situación real de la asignatura de Ingeniería de Software 1, en función de los principales problemas de los estudiantes en el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología de desarrollo de software.

Análisis documental: para el estudio de la documentación asociada al desarrollo de habilidades en la asignatura de Ingeniería de Software 1.

## 2.2 Resultados

Existen varios investigadores que han abordado el sistema como resultado científico. Donde el autor asume la definición: "El conjunto de componentes interrelacionados y lógicamente estructuradas que permiten la realización de un determinado trabajo profesional sobre la base de cumplir ciertas funciones y con el fin de lograr los objetivos trazados" [8].

La impartición de la asignatura, se basa en un aprendizaje creativo por parte de los alumnos, permitiendo una mayor versatilidad en la aplicación de los conocimientos, sin descuidar la necesidad de acercarse a los estudiantes en alguna medida a nuestras características y condiciones en el ámbito de desarrollo de software.

El enfoque de la asignatura debe ser eminentemente práctico de forma tal que el estudiante desarrolle las habilidades correspondientes en cada uno de los temas de la asignatura y los expresados en los objetivos generales de la misma, aunque no pueden faltar los elementos teóricos que ayudan a este desarrollo.

Por tanto la asignatura debe profundizar en aquellos aspectos de la vida profesional que el estudiante deberá enfrentar a corto plazo. Para lograr esto, se utilizará el estudio de casos extraídos de la vida real o preparada de forma similar. Este caso de estudio proporcionará al estudiante una ejercitación continua en relación con las distintas situaciones que enfrentará en el desarrollo de un software. De la misma forma se utilizará ejercicios teóricos que enriquecerá la parte práctica en los estudiantes.

Complementando los casos se deben preparar ejercicios y preguntas que contribuyan al logro de

los objetivos de la asignatura de manera individual y a distancia de manera muy sencilla.

## 2.3 Etapas del compendio de ejercicios

Teniendo en cuenta los elementos anteriores, el compendio de ejercicios está organizada en un conjunto de etapas que incluyen una serie de acciones interrelacionadas entre sí. Su objetivo es contribuir al desarrollo de la habilidad de seleccionar enfoque, modelo(s) y metodología en el proceso de desarrollo de software. Está dividida en cuatro etapas, que se muestran en la Figura 1.



**Figura. 1: Etapas del compendio de ejercicios.**  
Fuente: Elaboración propia

**1ra etapa. Diagnóstico:** Su objetivo es evaluar las habilidades primarias que los estudiantes necesitan para el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología a utilizar en el ciclo de vida de software.

Procedimientos

- Analizar y seleccionar el tipo de preguntas a incluir en cada uno de las preguntas a realizar a los estudiantes, que permitan conocer los principales problemas en el desarrollo de la habilidad.
- Confeccionar la guía de preguntas del cuestionario para la recolección de la información.
- Analizar los datos cualitativos obtenidos.
- Analizar las habilidades primarias que los estudiantes necesitan para el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología a utilizar en el ciclo de vida de software.

**2da etapa. Planificación:** Su objetivo consiste en analizar y determinar todas las condiciones necesarias para la puesta del compendio de ejercicios para el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología en el desarrollo de software.

Procedimientos

- Determinar en las formas de organización de la enseñanza de la asignatura Ingeniería de Software I los aspectos fundamentales que deben ser incluir en el compendio de ejercicios a desarrollar.
- Establecer las actividades docentes encaminadas a la adquisición de la habilidad de seleccionar el enfoque, modelo(s) y metodología a utilizar en el ciclo de vida de software, teniendo en cuenta los momentos de orientación ejecución y control.
- Determinar los ejercicios que tributen al desarrollo de las habilidades primarias detectadas en la etapa anterior para lograr el desarrollo de la habilidad de seleccionar del enfoque, modelo(s) y metodología a utilizar en el ciclo de vida de software.
- Determinar los criterios evaluativos del desarrollo de la habilidad de selección del enfoque, modelo(s) y metodología a utilizar en el ciclo de vida de software.

**3ra etapa. Ejecución:** Tiene como objetivo poner en práctica la actividad relacionada con el compendio de ejercicios para el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología a utilizar en el ciclo de vida de software.

Procedimientos

- Ejecutar el compendio de ejercicios para el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología del proceso de desarrollo de software.
- Garantizar la sistematización de las acciones para la adquisición de la habilidad.

**4ta etapa. Evaluación:** Con el objetivo de comprobar la efectividad del compendio de ejercicios en su propósito de lograr la adquisición de la habilidad de seleccionar el enfoque, modelo(s) y metodología para el proceso de desarrollo de software.

Procedimientos

- Evaluar la realización de las acciones según criterios establecidos para desarrollar la habilidad.
- Analizar el logro de la habilidad de seleccionar el enfoque, modelo(s) y metodología para el proceso de desarrollo de software en los estudiantes.

## 2.4 Ejemplificación del compendio de ejercicios

## Etapa 1 Diagnóstico:

El cuestionario realizado a los estudiantes para conocer los principales problemas en el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología para el proceso de desarrollo de software en los estudiantes.

A partir de los resultados obtenidos del cuestionario se determinaron los principales factores que pueden tener influencia en el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología para el proceso de desarrollo de software. Las insuficiencias existentes en el proceso de enseñanza-aprendizaje, que se resumen en: desmotivación por la asignatura de Ingeniería de Software I, existen pocos ejercicios para la autopreparación del estudiante en el desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología en el proceso de desarrollo de software, se le dedica poco tiempo al desarrollo de esta habilidad en la asignatura.

De acuerdo con los principales problemas detectados en el cuestionario es de suma importancia que los estudiantes sean capaces de desarrollar la habilidad de seleccionar el enfoque, modelo(s) y metodología para el proceso de desarrollo de software a partir de las habilidades primarias enunciadas anteriormente.

## Etapa 2 Planificación:

Compendio de ejercicios que tributen al desarrollo de la habilidad.

Es importante resaltar que la asignatura dedica al desarrollo de esta habilidad una conferencia, un seminario y una clase práctica. Al ser insuficiente las actividades docentes de manera presencial para el desarrollo de la habilidad el compendio de ejercicios se realiza para la preparación de los estudiantes en el estudio independiente o autopreparación.

Se establece el compendio de ejercicios que tributan al desarrollo de la habilidad de seleccionar el enfoque, modelo(s) y metodología del proceso de desarrollo de software de acuerdo a las habilidades primarias detectadas en la etapa de diagnóstico.

A continuación se explica cómo orientar al estudiante el grupo de ejercicios para el desarrollo de la habilidad:

### Ejercicios para el desarrollo de la habilidad

#### 1. Diga V o F según corresponda. Justifique los Falsos.

- \_\_\_ El enfoque ágil se basa en la búsqueda de la satisfacción del usuario.
- \_\_\_ El enfoque tradicional centra su atención

en llevar una planificación y documentación exhaustiva de todo el proyecto.

- \_\_\_ Los métodos híbridos pretenden retomar las ventajas de los métodos ágiles y tradicionales.
- \_\_\_ El enfoque ágil se basa en la entrega temprana de software incremental.
- \_\_\_ En el enfoque ágil los equipos de proyectos son grandes y altamente motivados.
- \_\_\_ Una característica importante dentro del enfoque tradicional es el alto costo al implementar un cambio.

#### 2. Argumente las afirmaciones siguientes:

- La agilidad se puede aplicar en cualquier proceso de software. Sin embargo, para lograrlo es esencial que el proceso sea diseñado en una forma que permita al equipo del proyecto adaptar y coordinar las tareas.
- Todos los modelos de proceso del software se ajustan a las actividades genéricas del marco de trabajo, pero cada uno aplica una importancia diferente a estas actividades.

#### 3. Complete con la palabra correcta (iterativo, software, en cascada, enfoque, en espiral, secuencial, basado en prototipos, eventos, usuario )

- En el modelo \_\_\_\_\_ el cliente debe tener paciencia. Una versión que funcione de los programas estará disponible cuando el proyecto esté muy avanzado.
- El modelo de proceso incremental, al igual que la construcción de prototipos y otros enfoques evolutivos, es \_\_\_\_\_ por naturaleza.
- El modelo DRA es una adaptación a “alta velocidad” del modelo en cascada en el que se logra el desarrollo rápido mediante un \_\_\_\_\_ de construcción basado en componentes.
- Los modelos \_\_\_\_\_ son iterativos, los caracteriza la forma en que permiten que los ingenieros de software desarrollen versiones cada vez más completas del software.
- El diseño rápido se centra en una representación de aquellos aspectos del \_\_\_\_\_ que serán visibles para el cliente o el usuario final.
- Un proceso \_\_\_\_\_ se divide en un conjunto de actividades del marco de trabajo que define el equipo de ingeniería de software.

g. El modelo de proceso concurrente define una serie de \_\_\_\_\_ que dispararán transiciones de estado a estado para cada una de las actividades, acciones o tareas de la ingeniería de software.

**4. Argumente las afirmaciones siguientes**

- a. La selección del modelo de desarrollo de software está condicionada por las características y condiciones del proyecto.
- b. Se pueden combinar en un mismo proyecto varios modelos de desarrollo de software.

**5. Seleccione el concepto de metodología correcto. Marque con una x.**

- a. \_\_\_ Herramienta que utiliza un procedimiento. Se pueden utilizar una o varias técnicas.
- b. \_\_\_ Para realizar una técnica, podemos apoyarnos en las herramientas software que automatizan su aplicación.
- c. \_\_\_ Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software.
- d. \_\_\_ Actividades elementales en que se dividen en procesos.

**6. Marque con una x las características deseables de una metodología**

- a. \_\_\_ Existencia de reglas predefinidas
- b. \_\_\_ Cobertura parcial del ciclo de desarrollo
- c. \_\_\_ Verificaciones intermedias
- d. \_\_\_ Planificación y control
- e. \_\_\_ Comunicación escasa
- f. \_\_\_ Utilización sobre un proyecto
- g. \_\_\_ Fácil formación
- h. \_\_\_ Herramientas CASE
- i. \_\_\_ Actividades que mejoren el proceso de desarrollo
- j. \_\_\_ Soporte al mantenimiento
- k. \_\_\_ Soporte de la reutilización de software

**7. Diga si las siguientes metodologías son ágiles (A) o Pesadas (P)**

- a. \_\_\_ Nova Open UP
- b. \_\_\_ Open UP
- c. \_\_\_ AUP (Proceso Unificado Ágil )
- d. \_\_\_ Métrica V3
- e. \_\_\_ SCRUM.
- f. \_\_\_ Feature -Driven Development (FDD).
- g. \_\_\_ Open Source Development Software
- h. \_\_\_ Dynamic Systems Development Method (DSDM).

- i. \_\_\_ Adaptive Software Development (ASD).
- j. \_\_\_ SXP
- k. \_\_\_ RUP (Rational Unified Process)
- l. \_\_\_ Extreme Programming (XP).
- m. \_\_\_ Crystal Clear.

**8. Justifique las afirmaciones siguientes del manifiesto ágil.**

- a. Valorar al individuo y a las interacciones del equipo de desarrollo por encima del proceso y las herramientas.
- b. Valorar el desarrollo de software que funcione por sobre una documentación exhaustiva.
- c. Valorar la colaboración con el cliente por sobre la negociación establecida.
- d. Valorar la respuesta al cambio por sobre el seguimiento de un plan.

**9. Enlace la columna A con la B con respecto a los diferentes enfoques y características de cada uno.**

Enfoques	Características
Enfoque tradicional	Provoca que los productos de software estén en cambios continuos y deben realizarse en forma acelerada. (ágil)
Enfoque ágil	Se mantiene la documentación exhaustiva y se incrementa el mantenimiento con entregas continuas (híbrido)
Enfoque híbrido	Cumplir con un plan de proyecto, definido en fases tempranas del proceso de desarrollo. (tradicional)

**10. Realice una comparación entre los diferentes enfoques de acuerdo a las características planteadas.**

Características	Ágil	Tradicional
Metas		
Tamaño		
Relaciones con el cliente		
Planificación y control		
Comunicación		
Requisitos		
Desarrollo		
Pruebas		
Cultura		

**11. Haga una comparación entre las**

**metodología tradicionales y ágiles de acuerdo a las características que se plantean a continuación**

Diferencias	Ágil	Tradicional
Cambios		
Cliente		
Arquitectura		
Contrato		
Entregas		
Documentación		

**12. Dada la problemática siguiente, identifique el enfoque, modelo y metodología a utilizar y argumente su selección.**

- a. Para el desarrollo de la aplicación se confeccionó un equipo compuesto por 30 miembros con la siguiente distribución: 8 analistas, 7 desarrolladores, 2 diseñadores de bases de datos, 1 jefe de proyecto, 2 planificadores, 4 aseguradores de la calidad del software y 6 estudiantes. Debido a que el cliente tiene muchas responsabilidades y la agenda de trabajo es muy apretada, se conciliaron encuentros mensuales para realizar chequeo de avance del proyecto. Se acordó, además, que es necesario que exista una documentación del proyecto que permita su mantenimiento por nuevos miembros. De los miembros del proyecto se conoce que todos tienen experiencia en el desarrollo de software de gestión. Ya han sido probados en ambientes de trabajo bajo presión y sus habilidades técnicas están experimentadas. Aunque se conoce lo que se debe realizar a grandes rasgos el equipo está preparado para que pueda enfrentar posibles cambios en las necesidades del cliente.
- b. OSI es un equipo de desarrollo de software formado por 15 personas, que en su mayoría se dedica a tiempo completo a esta tarea. Los clientes están a toda disposición del equipo de desarrollo formando parte del mismo. Quieren desarrollar un software de gran magnitud, en el que no se tienen claramente definidas todas las áreas que abarcará esa empresa. Los desarrolladores no están muy familiarizados con este modelo de sistema que pretenden obtener, pero se encuentran motivados y son expertos en el desarrollo basado en aspectos. Necesitarán muchos recursos por lo que pretenden que los riesgos de fallo sean mínimos. El cliente necesita una versión del producto dentro de

un mes.

- c. Se quiere desarrollar un portal web para el Ministerio de Trabajo y Seguridad Social. El cliente tiene bien definido las características del producto que desea, así como la tecnología con la que cuenta para utilizar la aplicación. Se prevé que sea un sistema de tamaño normal, con poca complejidad según los procesos que se llevan a cabo en la empresa. Dado estas características, la empresa contratada para el desarrollo de la aplicación seleccionó a un especialista de varios años en procesos de desarrollo de software y le asignó un equipo de 8 ingenieros recién graduados, dando como plazo 2 meses para la terminación del proyecto. El cliente está bastante comprometido con el proyecto por lo que no se observan inicialmente riesgos importantes que requieran de acciones por parte del equipo de desarrollo.
- d. Una empresa desarrolladora de software desea desarrollar un juego de simulación que permita enseñar a manejar. Esta idea surgió como propuesta del Dpto. de Tránsito a razón de una encuesta realizada en la autoescuela. Se han identificado además otras empresas desarrolladoras de software que están desarrollando aplicaciones muy similares a esta, por lo que se designó un equipo de 10 especialistas que conocen bien las tecnologías asociadas a los software de tiempo real y con varios años de experiencia como equipo. Se le dio al equipo un plazo de 2 meses para entregar una primera versión del producto para sacar al mercado, teniendo como condición que si se logra comercializar el mismo se mantendrá abierto el proyecto por un año y se incorporarán más especialistas al equipo inicial.

**Determinar los criterios evaluativos del desarrollo de la habilidad de selección del enfoque, modelo(s) y metodología a utilizar en el ciclo de vida de software.**

Para la evaluación del desarrollo de la habilidad de seleccionar enfoque, modelo(s) y metodología del proceso de desarrollo de software se propone realizarla en el corte del trabajo de curso en la semana 6 y en la prueba parcial en la semana 9 de la asignatura.

**Etapa 3 Ejecución**

La ejecución del compendio de ejercicios se debe llevarse a cabo según los tres momentos básicos de la actividad: orientación, ejecución y evaluación.

#### **Etapas 4 Evaluación**

La evaluación del compendio de ejercicios se realizará a través de la valoración del grado de adquisición de la habilidad de seleccionar enfoque, modelo(s) y metodología del proceso de desarrollo de software en los estudiantes puede ser obtenida a partir de los análisis cuanti-cualitativo de los resultados del corte del trabajo de curso 1 en la semana 6 del curso y en la promoción alcanzado en la prueba parcial de la asignatura y del correspondiente análisis que se efectúe en los estudiantes y colectivos de la asignatura.

#### **Recomendaciones para su instrumentación**

La instrumentación de este compendio de ejercicios debe ir precedida por su presentación y discusión con los profesores de la disciplina de Ingeniería de Software I, para que posteriormente este proceso sea reproducido en los colectivos de la asignatura de las diferentes facultades de forma tal que se logre la preparación del colectivo de profesores en los diferentes aspectos que componen este compendio de ejercicios; estas discusiones pueden servir para enriquecerla a través de los aportes de los profesores.

En relación con este aspecto se considera que resulta indispensable que el proceso de enseñanza-aprendizaje relacionado con la realización de la discusión diagnóstica, constituya una línea priorizada a tratar en el sistema de trabajo metodológico de los colectivos de asignatura a todos los niveles de las facultades.

Además resulta de vital importancia cuando se analice en los colectivos el sistema evaluativo a aplicar en la asignatura, que se tomen en cuenta los elementos expuestos anteriormente acerca de cómo abordar la evaluación de esta habilidad por los estudiantes.

#### **3. CONCLUSIONES**

Se diseñó un compendio de ejercicios en base a las deficiencias detectadas en un estudio previo con estudiantes y profesores de la asignatura de Ingeniería de Software I en la Facultad 1.

Su novedad científica está relacionada con el hecho de que no existen referencias anteriores de un

compendio de ejercicios similar en la que se abarquen aspectos didácticos relacionados con la adquisición de la habilidad de seleccionar enfoque, modelo(s) y metodología del proceso de desarrollo de software, expresados en las acciones a desarrollar por los estudiantes para la ejecución de esta actividad.

#### **4. REFERENCIAS BIBLIOGRÁFICAS**

1. Pérez, L. Modelado de negocio y gestión de requisitos como etapas imprescindibles para el desarrollo de los sistemas automatizados de información, 2008.
2. Pressman, R.S. Ingeniería del Software. Un enfoque práctico, ed. 6ta. 2007.
3. Lynch, J., Standish Group - Chaos report 2012.
4. UCI, C.d.C.d.I., PLAN DE ESTUDIOS "D" INGENIERÍA EN CIENCIAS INFORMÁTICAS. 2014, Universidad de las Ciencias Informáticas. p. 124.
5. Informáticas, U.d.I.C. 2015; Available from: [www.uci.cu](http://www.uci.cu).
6. Tejera, I.D.C.H., Informe Semestral. 2010-2011. p. 6.
7. Tejera, M.D.C.H., PROGRAMA ANALÍTICO. 2015: p. 43.
8. Collazo, Z.S.L., Sistema de entrenamientos para la formación permanente de los profesores de la especialidad de agronomía del instituto politécnico "Protesta de Baraguá.. IPLAC, 2012.