



Identificación del curso

Nombre del Ova o curso:	Ingeniería del software
Programa:	Ingeniería del Software
Escritor:	Oralia Cortés G
Año versión 1	2014 Heber Gómez
Año, versión 2	2016 Oralia cortés Grajales
Año, versión 3	2017 Oralia cortés Grajales
Número de créditos:	4
Competencia Global del curso:	Aplicar alternativas actualizadas y avanzadas en el desarrollo y la entrega de productos de software de calidad, los cuales satisfagan plenamente los requerimientos de los clientes.

Estructura

Elemento de competencia 1: Aplicar los conocimientos adquiridos, para el desarrollo, gestión, mantenimiento, pruebas y entrega de los requisitos del usuario final, implementados en un sistema informático de manera eficiente, fiable, asequible y con calidad.	
TEMAS	HORAS

	AC: 16	TI: 48
Conceptos básicos de la ingeniería del software.	3	9
Definiciones, problemática y retos de la ingeniería del software.	3	9
Gestión de Proyectos.	2	6

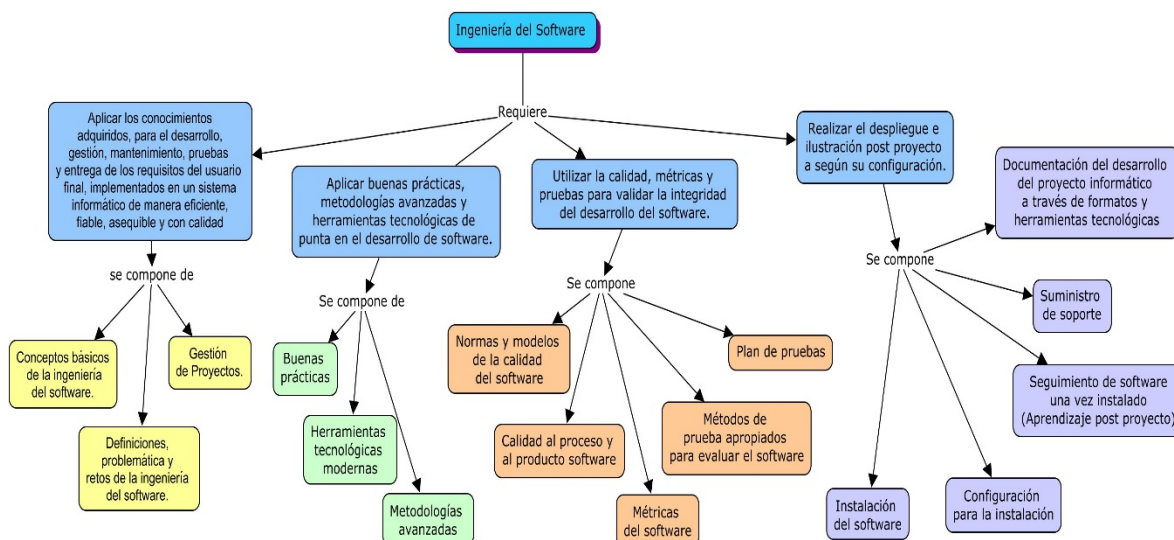
Elemento de competencia 2: Aplicar buenas prácticas, metodologías avanzadas y herramientas tecnológicas de punta en el desarrollo de software.		
TEMAS	HORAS	
	AC: 16	TI: 48
Buenas prácticas	3	9
Metodologías avanzadas	3	9
Herramientas tecnológicas modernas	2	6

Elemento de competencia 3: Utilizar la calidad, métricas y pruebas para validar la integridad del desarrollo del software.		
TEMAS	HORAS	
	AC: 16	TI: 48
Normas y modelos de la calidad del software.	4	12
Calidad al proceso y al producto software	2	6
Métricas del software	4	12
Métodos de prueba apropiados para evaluar el software.	4	12
Plan de pruebas	2	6

Elemento de competencia 4: Realizar el despliegue e ilustración post proyecto a según su configuración.		
TEMAS	HORAS	
	AC: 16	TI: 48

Configuración para la instalación.	4	12
Instalación del software.	4	12
Suministro de soporte.	2	6
Seguimiento de software una vez instalado (Aprendizaje post proyecto).	2	6
Documentación del desarrollo del proyecto informático a través de formatos y herramientas tecnológicas.	4	12

Mapa Conceptual



Metodología

El desarrollo del presente modulo promueve el aprendizaje de tipo investigativo, recursivo, encaminado a la asimilación y ejecución práctica de los conocimientos y saberes adquiridos, se utiliza una metodología basada en problemas donde el estudiante juega un papel importante en el proceso de aprendizaje en la solución a la situación problemática planteada, el aprendizaje basado en problemas que tiene como premisa fundamental que la enseñanza debe iniciarse con el planteamiento de

problemas que sean de importancia y pertinencia para los estudiantes. Una teoría de diseño de este tipo de aprendizaje debe tener orientaciones que ayuden a secuencias de problemas contruidos unos sobre otros, en orden de complejidad. Cuando un problema finalmente se resuelve, los estudiantes aprenden volviendo al ámbito del problema, para replantearlo, evaluar su solución, recorrer las secuencias, en general, se determina un proceso de razonamiento posterior altamente constructivo.

Se plantea la continua necesidad de exploración de fuentes y recursos, actualización y continua investigación dentro de cada uno de los ejes temáticos propuestos en el módulo, el presente modulo cuenta el empleo de materiales de aprendizaje de alta calidad, los cuales han sido diseñados o adecuados para facilitar procesos de aprendizaje del estudiante en la modalidad virtual, aparte que se considera la temática de la ingeniería del software fundamental en el desarrollo de software.

Se hace vital y necesaria la continua interacción, comunicación y complementación entre el tutor y el estudiante, mediante los comentarios, aportes, preguntas aprovechando las herramientas virtuales de comunicación, a fin de recibir la oportuna orientación y complementación dentro del desarrollo de la situación o caso propuesto.

El estudiante es el centro del proceso y su disciplina y autoestudio se constituyen en factores fundamentales para esta modalidad educativa. Adicionalmente las herramientas de la plataforma ofrecen espacios para la presentación de los contenidos, la interacción, la entrega de evidencias y verificación de competencias alcanzadas.

Las actividades de aprendizaje propuestas, permiten dejar plasmada el logro de las competencias, gracias a la solución de problemas de las situaciones propuestas que se realizaran en entregas parciales, foros de discusión, video tutoriales y tutorías sincrónicas durante el curso.

Evaluación

Para llegar al logro de competencias propuestas, es necesario que el estudiante asimile los materiales propuestos en el módulo para conocer, comprender, aplicar, analizar, sintetizar, evaluar e innovar en la solución de problemas de las situaciones propuestas que se realizaran en entregas parciales, temas de discusión en foros o wikis, autoevaluación en el portafolio y tutorías sincrónicas durante el curso.

La evaluación por competencias en la Católica del Norte se orienta por los siguientes principios:

1. Continua: no hay momentos específicos para la evaluación, se realiza durante todo el periodo y a través de las actividades programadas en los módulos.
2. Integral: se valoran el ser, el saber y el hacer; es decir, el ser humano en todas sus dimensiones.
3. Refuerzo permanente: está implícito en el desarrollo de las actividades, porque el mismo criterio de desempeño se puede valorar en diferentes momentos. Los refuerzos culminan a la par de la terminación del bloque.

La evaluación de las competencias se realiza a partir de la entrega de las evidencias, éstas a su vez están planteadas como actividades y cada una cuenta con una serie de criterios que estandarizan la valoración de los resultados.

Los criterios de desempeño definidos para cada elemento de la competencia, son la base para determinar los resultados de aprendizaje que se estructuran con base en EVIDENCIAS DE APRENDIZAJE que son las pruebas manifiestas de aprendizaje, recogidas directamente durante el proceso formativo. Son recolectadas con la orientación del facilitador, utilizando técnicas, métodos e instrumentos de evaluación seleccionados, según sean evidencias de conocimiento, de producto o de desempeño, permitiendo reconocer los logros obtenidos por el estudiante en los tres tipos de saberes: conceptual, procedimental y actitudinal.

EVIDENCIAS DE CONOCIMIENTO. Apuntan al dominio cognoscitivo para procesar e identificar información relevante, su clasificación, su interpretación de manera útil, y la búsqueda de las relaciones entre información nueva e información adquirida previamente. Incluye el conocimiento de hechos y procesos, la comprensión de los principios, y teorías y las maneras de utilizar el conocimiento en situaciones cotidianas y nuevas.

EVIDENCIAS DE DESEMPEÑO. Evidencias del saber procedimental, relativas al cómo ejecuta el estudiante una actividad, en donde pone en juego sus habilidades, conocimientos y actitudes. Permiten recoger información directa, de mejor calidad y más confiable, sobre la forma como el estudiante desarrolla su proceso de aprendizaje y así poder identificar cuáles han sido sus logros y cuáles le faltan por alcanzar. Incluye las evidencias actitudinales.

EVIDENCIAS DE PRODUCTO. Son los resultados que obtiene el estudiante en una actividad que refleja el aprendizaje alcanzado y permite hacer inferencias sobre el proceso desarrollado, o método utilizado.

Justificación

La ingeniería de software es un área del conocimiento relativamente joven, lo cual implica que sea difícil afrontar muchos tipos de problemas que no se han tratado antes, y lo que es peor, que muchos proyectos de software no culminen como se espera. De acuerdo con una encuesta del año 2004 realizada por Standish Group, solamente el 29% de los proyectos de software son exitosos y se realizan bajo los recursos estimados; la misma encuesta dice que el restante 71% son proyectos desviados o cancelados.

De igual manera, es importante tener en cuenta que el software hace cada vez más parte de nuestras vidas; y su relevancia juega cada vez un papel más crítico en la sociedad. Hoy en día no solamente se requiere que el software se entregue a los clientes dentro del tiempo pactado, sino que éste se caracterice por un conjunto de aspectos de calidad que garanticen la satisfacción de los clientes; y más significativo aún, que no ponga en riesgo el curso de un negocio o las personas que utilizan el software.

Por las razones previamente expuestas, las empresas de desarrollo de software gastan gran parte del tiempo y del presupuesto en asegurar la calidad del software. Estas actividades se enfocan en mejorar tanto el proceso de desarrollo como el producto final y los artefactos intermedios.

Durante el curso “Ingeniería de Software”, se van a presentar los fundamentos, conceptos y utilización más relevantes de buenas prácticas, metodologías avanzadas y herramientas tecnológicas de punta en el desarrollo de software; igualmente se van a presentar la calidad, métricas y pruebas para validar la integridad del desarrollo del software, con el propósito que los estudiantes aprendan a aplicarlas dentro de un contexto real y adquieran habilidades analíticas y constructivas para que el desarrollo de software sea de calidad.

Desarrollar un software funcional no es una tarea compleja pero cuando éste debe incluir aspectos de alta disponibilidad, seguridad, rendimiento, confiabilidad, entre otros, es cuando se convierte en una tarea difícil y se puede convertir en imposible sino se aborda desde la arquitectura y el diseño de software para que se lleve al despliegue del sistema e ilustración post proyecto a según su configuración.

Glosario

CAD: Computer-Aided Design (Diseño asistido por computador)

CASE: Computer Aided Software Engineering (Ingeniería de software asistida por computador)

CPF: Common Process Framework (Marco de proceso común)

DSDM: Dynamic Systems Development Method (Método de desarrollo de sistemas dinámico)

IEC: International Electrotechnical Commission

IEEE: Institute of Electrical and Electronics Engineers

ISO: International Organization for Standardization

UP: Unified Process (Proceso Unificado)

RAD: Rapid Application Development (Desarrollo Rápido de Aplicaciones)

RUP: Rational Unified Process (Proceso Unificado Rational)

SDLC: Software Development Life Cycle (Ciclo de vida de desarrollo de software)

TDD: Test Driven Development (Desarrollo orientado a pruebas)

UML: Unified Modeling Language

XP: Extreme Programming (Programación extrema)

Aseguramiento de la calidad del software (SQA): es un conjunto de actividades planificadas y ejecutadas sistemáticamente que apunta a asegurar que el software que se está construyendo es de alta calidad. Recuperado de:
https://www.incibe.es/file/N85W1ZWFHifRgUc_oY8_Xg

Ciclo de vida del software: El conjunto de procesos sistemáticos que tienen lugar durante la existencia del producto, desde su concepción inicial hasta que la organización decide no continuar manteniéndolo. Recuperado de:
<http://reocities.com/siliconvalley/lab/3629/glosario.htm>

CMMI (*Modelo integrado de madurez de la capacidad*) se usa para evaluar el nivel de madurez de una compañía en términos de desarrollo informático. Recuperado de:

<http://es.ccm.net/contents/595-cmmi-modelo-integrado-de-madurez-de-la-capacidad>

Medición (*measurement*) [por L. Fernández Sanz]: Proceso por el cual se asignan números o símbolos a atributos de entidades del mundo real de tal forma que los describa de acuerdo con reglas claramente definidas [Fenton y Pfleeger, 1997, p. 5]. Nota: las definiciones estandarizadas alternativas son más restrictivas y no recogen la teoría básica de la medición. [Por L. Fernández Sanz]

Recuperado de: <http://www.sc.ehu.es/jiwdocoj/remis/docs/terminos.html>

Métrica (*metric*) [por L. Fernández Sanz]: medida cuantitativa del grado en el que un sistema, un componente o un proceso posee un atributo dado [IEEE, 1990, p. 130]. Nota: esta definición estandarizada no contempla la aplicación de métricas a otras entidades que no sean productos o procesos (p.ej., recursos, proyectos, cliente, etc.).

Recuperado de: <http://www.sc.ehu.es/jiwdocoj/remis/docs/terminos.html>

Métodos de ingeniería del software: los métodos de ingeniería del software proporcionan el “cómo” para la construcción del software. Abarcan las tareas técnicas requeridas para realizar y documentar el análisis de requisitos, el diseño, la construcción de programas, las pruebas y el mantenimiento. Recuperado de: https://www.incibe.es/file/N85W1ZWFHifRgUc_oY8_Xg

Modelos de proceso ágiles: los modelos de proceso ágiles son enfoques de desarrollo donde los requisitos del cliente se cumplen temprano en el ciclo de vida de desarrollo del software a través de continuas entregas de software. En estos modelos, los cambios en los requisitos son bienvenidos. Recuperado de: https://www.incibe.es/file/N85W1ZWFHifRgUc_oY8_Xg

Modelo de proceso de software: un modelo de proceso de software es una estrategia abstracta para la construcción y mantenimiento de un producto software.

Ayuda a los jefes de proyecto en la planificación y ejecución de un proyecto. También se le llama modelo de ciclo de vida o paradigma de ingeniería del software.

Recuperado de: https://www.incibe.es/file/N85W1ZWFHifRgUc_oY8_Xg

Modelo de proceso incremental: el modelo de proceso es un enfoque en el que el producto se entrega al cliente incrementalmente sobre un periodo de tiempo planificado. Recuperado de: https://www.incibe.es/file/N85W1ZWFHifRgUc_oY8_Xg

Proceso: secuencia de pasos para realizar alguna actividad e incluye la descripción de entradas, salidas, procedimientos, herramientas, responsabilidades y criterios de salida. Recuperado de: https://www.incibe.es/file/N85W1ZWFHifRgUc_oY8_Xg

Programación extrema, o Extreme Programming (XP): es una metodología de desarrollo ágil, una de las más exitosas en tiempo reciente. Su autor principal es Kent Beck, quien eligió algunas características de otras metodologías y las relacionó de forma que cada una complementara a la otra. Recuperado de: http://www.uv.mx/universo/486/infgral/infgral_15.html

SCRUM: es una metodología ágil y flexible que sirve para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa. Se basa principalmente en construir la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación. Recuperado de: <http://www.sc.ehu.es/jiwdocoj/remis/docs/terminos.html>