Universidad Autónoma del Estado de México
Unidad Académica Profesional Tianguistenco
Licenciatura en Ingeniería de Software

Guía pedagógica:

Métodos y modelos de desarrollo de software

Elaboró: LSCA Carlos Alberto García Acevedo

Fecha: 30/junio/2015

Fecha de aprobación	H. Consejo académico	H. Consejo de Gobierno

Índice

	Pág.
I. Datos de identificación	3
II. Presentación de la guía pedagógica	4
III. Ubicación de la unidad de aprendizaje en el mapa curricular	5
IV. Objetivos de la formación profesional	5
V. Objetivos de la unidad de aprendizaje	6
VI. Contenidos de la unidad de aprendizaje, y su organización	6
VII. Acervo bibliográfico	12
VIII. Mapa curricular	13

I. Datos de identificación

Espacio educativo donde se imparte	**Unidad Académica Profesional Tianguistenco**			
Licenciatura	**Licenciatura en Ingeniería de Software**			
Unidad de aprendizaje	**Métodos y modelos de desarrollo de software**	**Clave**	**L40817**	
Carga académica	**3** Horas teóricas	**2** Horas prácticas	**5** Total de horas	**8** Créditos

Período escolar en que se ubica

1	2	3	4	**5**	6	7	8	9

Seriación	Requisitos y especificación de software	Ninguna
	UA Antecedente	UA Consecuente

Tipo de Unidad de Aprendizaje

Curso		Curso taller	✓
Seminario		Taller	
Laboratorio		Práctica profesional	
Otro tipo (especificar)			

Modalidad educativa

Escolarizada. Sistema rígido		No escolarizada. Sistema virtual	
Escolarizada. Sistema flexible	✓	No escolarizada. Sistema a distancia	
No escolarizada. Sistema abierto		Mixta (especificar)	

Formación común

Formación equivalente	**Unidad de Aprendizaje**

II. Presentación de la guía pedagógica

Es importante que el docente cuente con una Guía pedagógica para realizar su actividad, la cual tiene como propósito complementar el programa de estudios y al mismo tiempo orientar el proceso y ejecución de su intervención educativa, permitiéndole reflexionar sobre las actividades que va a proponer a los alumnos, los medios didácticos que se utilizará, proponer prácticas y solucionar problemas que se presentan en la utilización de las computadoras y el software utilizados con fines educativos durante las sesiones de aprendizaje.

En la presente Guía pedagógica de la Unidad de Aprendizaje Métodos y modelos de desarrollo de software se presenta una propuesta didáctica que permitirán al docente desarrollar su programa y enriquecerlo con nuevos conocimientos y actualizaciones, en congruencia con el modelo de competencias y la enseñanza constructivista, a fin de contribuir a la formación de los profesionales de la Carrera de Ingeniería de Software, atendiendo a sus particulares estilos de aprendizaje.

Esta unidad de aprendizaje comprende conocer e identificar los diferentes fundamentos asociados a los métodos y modelos de desarrollo de software así como establecer criterios para aplicar el patrón de diseño más adecuado en cada caso de construcción, mejora o actualización, fundamentándose en los sistemas basados en componentes establecidos por la ingeniería de software. El entorno de desarrollo de la Unidad de aprendizaje es en el aula y en laboratorio.

Para lograr lo anterior el estudiante debe conocer conceptos y fundamentos de Ingeniería de software, así como contar con técnicas para el análisis y el diseño. También debe ser capaz de aplicar, distinguir y proponer adecuadamente las técnicas para obtener requisitos y especificaciones enfocados a solucionar problemas prácticos así como tener la habilidad de emplearlos para crear, mejorar o actualizar software con una actitud de valoración de la contribución de las diversas disciplinas de desarrollo de programas que buscan simplificar el trabajo humano.

La unidad de aprendizaje pertenece al área curricular Tratamiento de información, núcleo promueve una relación inter y trans-disciplinaria al agrupar unidades de aprendizaje que definen fundamentalmente la formación práctica o aplicativa de la profesión y el desarrollo de competencias específicas. También provee al alumno de escenarios educativos para la integración, aplicación y desarrollo de los conocimientos, habilidades y actitudes que le permitan el desempeño de las funciones, tareas y resultados requeridos en los diferentes ámbitos de intervención profesional.

Es importante el empeño e interés del alumno para dominar y adquirir habilidades en el manejo de la computadora y del software que se utilizarán en la unidad de aprendizaje, así como las propuestas para integrar conocimientos técnicos, prácticos, disciplinarios, pedagógicos o didácticos en el proceso y ejecución del enriquecimiento educativo, de esta manera el docente, facilitador del aprendizaje, podrá construir programas de estudios de una unidad de aprendizaje por objetivos académicos y profesionales.

El estudiante amplía su visión al percibir cómo la industria está actualmente empleando diversas técnicas para desarrollar software en un entorno tan cambiante y exigente. Al mismo tiempo valora y aplica los conocimientos y habilidades adquiridos en su carrera hasta este punto para resolver diferentes situaciones prácticas mientras va haciéndose más consciente del relevante papel que desempeñan hoy los métodos y modelos en el desarrollo, así como la misma Ingeniería de software.

III. Ubicación de la unidad de aprendizaje en el mapa curricular

Núcleo de formación:	Integral
Área Curricular:	Tratamiento de Información
Carácter de la UA:	Obligatoria

IV. Objetivos de la formación profesional.

Objetivos del programa educativo:

Formar profesionistas con los conocimientos, habilidades y actitudes necesarios para contribuir en cualquiera de los procesos de la Ingeniería de Software para proponer soluciones de calidad al manejo automatizado de información dentro de las organizaciones, aplicando un enfoque sistemático, disciplinado y cuantificado en la formulación, planeación, análisis, diseño, implantación y mantenimiento de software, así como la generación de conocimiento, metodologías y métricas en torno a la Ingeniería de Software .

Objetivos del núcleo de formación:

Núcleo de formación Integral.

Proveer al alumno/a de escenarios educativos para la integración, aplicación y desarrollo de los conocimientos, habilidades y actitudes que le permitan el desempeño de las funciones, tareas y resultados ligados directamente a las dimensiones y ámbitos de intervención profesional o campos emergentes de la misma.

Objetivos del área curricular o disciplinaria:

Tratamiento de información:

Adquirir los conocimientos necesarios para el diseño y realización de sistemas de bases de datos, considerando aspectos de análisis, organización lógica y física, determinación del modelo apropiado, así como selección y aplicación de las herramientas adecuadas tomando en cuenta los principios de las bases de datos y sus diferentes modelos.

Contar con los elementos teóricos requeridos para el manejo y recuperación de grandes volúmenes de información.

Aplicar las distintas teorías, técnicas y metodologías de análisis y diseño para la concepción y entendimiento de sistemas de manejo de información, para modelar situaciones del entorno real, resolver problemas y optimizar la toma de decisiones.

V. Objetivos de la unidad de aprendizaje.

Analizar y aplicar los diferentes métodos de desarrollo de software e identificar el más adecuado para una implementación particular, fundamentando en los modelos existentes derivados de la ingeniería de software.

VI. Contenidos de la unidad de aprendizaje, y su organización.

<table>
<tr><td colspan="3">Unidad 1. Necesidad y surgimiento de métodos y modelos para el desarrollo de software.</td></tr>
<tr><td colspan="3">Objetivo:
Valorar la contribución que han hecho tanto la Ingeniería de software, así como las principales disciplinas que han propuesto los métodos y modelos existentes para crear, mejorar o actualizar software, comprendiendo e identificando los requerimientos específicos y reconociendo los rasgos que los categorizan.</td></tr>
<tr><td colspan="3">Contenidos:
1.1 Necesidad de métodos y modelos para el desarrollo de software.
1.2 Principales disciplinas de desarrollo y soporte.
1.3 Contribución de la ingeniería de software.</td></tr>
<tr><td colspan="3">Métodos, estrategias y recursos educativos</td></tr>
<tr><td colspan="3">Discusión y análisis, lluvia de ideas, trabajo en equipos, mapa conceptual, cuadro sinóptico, método de caso, ABP, síntesis, exposición y clase magistral.</td></tr>
<tr><td colspan="3">Actividades de enseñanza y de aprendizaje</td></tr>
<tr><td>Inicio</td><td>Desarrollo</td><td>Cierre</td></tr>
<tr><td>1.1. Lluvia de ideas e investigación personal. Elabora un diagrama con los principales elementos de la crisis del software.</td><td>1.1. Clase magistral. Anotación de los puntos más relevantes expuestos de la necesidad de la industria del software de contar con estándares y modelos de desarrollo. Entrega síntesis.</td><td>1.1. En equipos de trabajo elabora mapa conceptual de la necesidad inicial de la industria de definir métodos y modelos para el desarrollo de software de calidad.</td></tr>
<tr><td>1.2. Método de caso. Discusión en grupos pequeños sobre las principales disciplinas que auxiliaron con técnicas a la industria. Entrega reporte.</td><td>1.2. Debate donde se analizan y proponen diferentes puntos de vista sobre las principales disciplinas que apoyaron a la industria para el desarrollo y soporte. Entrega conclusiones.</td><td>1.2. Elabora cuadro sinóptico sobre cómo aportaron a la creación, actualización y soporte las principales disciplinas de desarrollo.</td></tr>
</table>

1.3. Clase magistral. Anotación de los puntos más relevantes expuestos de las principales características de la ingeniería de software. Entrega resumen.	1.3. Aprendizaje basado en problemas. Mediante investigación y esquemas el alumno identificará el papel y la importancia de la ingeniería de software como disciplina para desarrollar adecuadamente	1.3. En equipos de trabajo. Exposición y reporte donde se valora el papel que desempeñan hoy los métodos, modelos y estándares en la adecuada aplicación de la Ingeniería de software.
(2 Hrs.)	**(7 Hrs.)**	**(3 Hrs.)**
Escenarios y recursos para el aprendizaje (uso del alumno)		
Escenarios	**Recursos**	
Aula	Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.	

Unidad 2. Metodologías de desarrollo de software.		
Objetivo: Analizar las características, ventajas y desventajas propias de cada método de desarrollo que permita determinar los procedimientos y ámbitos de aplicación más convenientes para los distintos casos y contextos donde surja la necesidad de seleccionar o recomendar una metodología.		
Contenidos: **2.1 Ciclo de vida en cascada o secuencial** **2.2 Desarrollo por análisis estructurado** **2.3 Desarrollo basado en componentes** **2.4 Proceso unificado** **2.5 Metodologías ágiles**		
Métodos, estrategias y recursos educativos		
Discusión y análisis, investigaciones, trabajo en equipos, cuadro comparativo, exposición y clase magistral.		
Actividades de enseñanza y de aprendizaje		
Inicio	**Desarrollo**	**Cierre**
2.1. Clase magistral. Anotación de los puntos más relevantes expuestos de las características de lo que son las metodologías de desarrollo de software. Presenta resumen.	2.1. Investigación de las características de la metodología secuencial y entrega un cuadro sinóptico con los resultados.	2.1. En equipos de trabajo. Exposición y reporte donde presentan ventajas, desventajas y usos de la metodología secuencial.

2.2., 2.3., 2.4. y 2.5. Clase magistral. Anotación de los puntos más relevantes expuestos de las características de las diferentes metodologías de desarrollo de software. Entrega resumen.	2.2., 2.3., 2.4. y 2.5. Investigación y análisis aportando diferentes puntos de vista sobre las características de cada una de las metodologías. Presenta esquema con las principales características.	2.2., 2.3., 2.4. y 2.5. Mediante un cuadro comparativo el alumno conoce e identifica las diferentes metodologías de desarrollo así como sus usos en la práctica.
(4 Hrs.)	**(10 Hrs.)**	**(4 Hrs.)**

Escenarios y recursos para el aprendizaje (uso del alumno)	
Escenarios	**Recursos**
Aula Laboratorio	Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.

Unidad 3. Modelos de desarrollo de software.		
Objetivo: Analizar los rasgos particulares, ventajas e inconvenientes de cada modelo de desarrollo así como las condiciones y requisitos establecidos por la industria del software con el fin de contar con los elementos necesarios para proponer y aplicar el modelo más adecuado, así como las metodologías con las que se pueden asociar para un consistente y adecuado desarrollo.		
Contenidos: **3.1 Modelo en cascada** **3.2 Modelo por etapas o procesos** **3.3 Modelo incremental** **3.4 Modelo espiral** **3.5 Modelo por prototipos** **3.6 Modelo orientado o objetos** **3.7 Modelo RAD**		
Métodos, estrategias y recursos educativos		
Discusión y análisis, investigaciones, trabajo en equipos, cuadro comparativo, exposición y clase magistral.		
Actividades de enseñanza y de aprendizaje		
Inicio	**Desarrollo**	**Cierre**
3.1. Clase magistral. Anotación de los puntos más relevantes expuestos de las características de lo que son los modelos de desarrollo de software. Presenta resumen.	3.1. Investigación de las características del modelo en cascada y entrega un cuadro sinóptico con los resultados.	3.1. En equipos de trabajo. Exposición y reporte donde presentan el alcance, los recursos y usos del modelo en cascada.

3.2., 3.3., 3.4., 3.5., 3.6. y 3.7. Clase magistral. Anotación de los puntos más relevantes expuestos de las características de los diferentes modelos de desarrollo de software. Entrega resumen.	3.2., 3.3., 3.4., 3.5., 3.6. y 3.7. Investigación y análisis aportando diferentes puntos de vista sobre las bases y características de cada uno de las modelos. Presenta esquema con los principales atributos.	3.2., 3.3., 3.4., 3.5., 3.6. y 3.7. Mediante un cuadro comparativo el alumno conoce e identifica los diferentes modelos de desarrollo así como sus adecuaciones en la práctica.
(4 Hrs.)	**(10 Hrs.)**	**(4 Hrs.)**

Escenarios y recursos para el aprendizaje (uso del alumno)

Escenarios	Recursos
Aula Laboratorio	Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.

Unidad 4. Técnicas de evaluación del desarrollo de software.

Objetivo:
Comprender la importancia de las técnicas, mediciones y estimaciones de calidad que se pueden aplicar al análisis y diseño, identificando los criterios más adecuados para proponer y colaborar con la evaluación del proceso de desarrollo, determinando su impacto tanto en el desarrollo como en el producto final de software, logrando optimizar el empleo de recursos.

Contenidos:

4.1 Importancia de la evaluación del análisis y el diseño antes de desarrollar
4.2 Técnicas, mediciones y estimaciones propias de la etapa de construcción del software
4.3 Pruebas y aseguramiento de la calidad en el proceso de desarrollo de software

Métodos, estrategias y recursos educativos

Discusión y análisis, lluvia de ideas, cuadro sinóptico, método de caso, ABP, Debate, trabajo en equipos, exposición y clase magistral.

Actividades de enseñanza y de aprendizaje

Inicio	Desarrollo	Cierre
4.1. Lluvia de ideas e investigación personal. Elabora un diagrama con los principales aspectos por los que es importante evaluar el análisis y el diseño previo al desarrollo de software.	4.1. Clase magistral. Anotación de los puntos más relevantes del efecto de evaluar bien o no las primeras etapas de desarrollo antes de proceder a construir el software. Entrega resumen.	4.1. En equipos de trabajo. Elabora mapa conceptual con los principales criterios de evaluación del análisis y el diseño para lograr el posterior desarrollo de software de calidad.

4.2. Método de caso. Discusión en grupos pequeños sobre las principales formas de medir y estimar la calidad en la etapa de construcción del software. Entrega reporte.	4.2. Clase magistral. Anotación de los puntos expuestos de las principales técnicas, mediciones y estimaciones aplicables a la etapa de construcción del software. Entrega resumen.	4.2. Elabora cuadro sinóptico con los criterios que ayudan a evaluar y medir la adecuada construcción del software.
4.3. Debate donde se analizan y proponen diferentes puntos de vista sobre la importancia de implementar o no pruebas de calidad en el proceso de desarrollo de software. Entrega conclusiones.	4.3. Aprendizaje basado en problemas. Mediante investigación y un esquema el alumno identifica los parámetros de calidad aplicables al proceso de desarrollo software.	4.3. En equipos de trabajo. Exposición y reporte donde se indican los beneficios de aplicar pruebas y de contar con aseguramiento de calidad en el todo proceso de desarrollo de software.
(2 Hrs.)	**(10 Hrs.)**	**(4 Hrs.)**

Escenarios y recursos para el aprendizaje (uso del alumno)	
Escenarios	**Recursos**
Aula Laboratorio	Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.

Unidad 5. Selección e implementación de la metodología y del modelo de desarrollo.
Objetivo: Aplicar con actitud de iniciativa, responsabilidad y asertividad las técnicas, estándares y criterios más convenientes a un caso de creación de software, seleccionando el método y el modelo pertinentes y, de acuerdo a los requerimientos de capacitación, soporte y costo, proponer la solución más adecuada para el proceso de desarrollo.
Contenidos: **5.1 Selección del método de desarrollo más adecuado** **5.2 Técnicas y criterios para implementar un modelo** **5.3 Selección e implementación de técnicas para desarrollar software específico**
Métodos, estrategias y recursos educativos
Taller, investigaciones, cuadro sinóptico, resolución de problemas y clase magistral.
Actividades de enseñanza y de aprendizaje

Inicio	**Desarrollo**	**Cierre**

5.1., y 5.2. En equipos de trabajo investiga las técnicas y criterios que se emplean por la industria para seleccionar e implementar un método y un modelo en particular para el ciclo de producción de un software específico. Entrega conclusiones.	5.1., y 5.2. Clase magistral. Anotación de los puntos más relevantes expuestos de los criterios para seleccionar el método para casos específicos así como para implementar un modelo de desarrollo, de acuerdo a sus características propias. Entrega resumen.	5.1., y 5.2. Elabora cuadro sinóptico con los principales criterios que ayudan a seleccionar e implementar el método y el modelo a casos de desarrollo de características específicas.
5.3. En equipos de trabajo seleccionan y proponen el caso y software para el que plantearán el plan de desarrollo. Entrega propuesta de desarrollo.	5.3. Taller supervisado para resolver el problema planteado en equipos de trabajo de acuerdo a los criterios indicados. Entrega avances.	5.3. Exposición y entrega de trabajo final con la propuesta completa para el desarrollo del software específico.
(1 Hrs.)	**(10 Hrs.)**	**(5 Hrs.)**

Escenarios y recursos para el aprendizaje (uso del alumno)	
Escenarios	**Recursos**
Aula Laboratorio	Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.

VII. Acervo bibliográfico

Básico:

1. Piattini, M. García, F. Garzás, J. & Genaro, M. (2008), *Medición y estimación del software: Técnicas y métodos para mejorar la calidad y la productividad (1a edición)*: Alfaomega/Ra-Ma Editorial. Madrid, España
2. Pressman, R. (2006), *Ingeniería de Software: Un enfoque Práctico (6a edición)*: McGraw-Hill Interamericana. México, D.F. México
3. Summerville, I. (2005), *Ingeniería de Software (7a edición)*: Pearson/Addison-Wesley. Madrid, España
4. Whitten, J. & Bentley, L. (2008), *Análisis de sistemas, Diseño y métodos (7a edición)*: McGraw-Hill Interamericana. México, D.F. México

Complementario:

1. Greenfield, J. Short, K. Cook, S. & Kent, S. (2003) *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*: ACM, ISNB: 1581137516. New York, NY, USA.
2. Jacky, J. (2001) *The Way of Z: Practical Programming with Formal Methods*: Cambridge University Press. USA.
3. Jacobson, I. Booch, G. & Rumbaugh, J. (2000) *El Proceso Unificado de Desarrollo de Software*: Pearson/Addison-Wesley. Madrid, España.
4. Kelly, S. & Tolvanen, J. (2008) *Domain-Specific Modeling: Enabling Full Code Generation*: John Wiley & Sons,Inc. New Jersey, USA
5. Pastor, O. & Molina, J. (2004) *Model driven architecture in practice*: Springer. Valencia, España.
6. Stahl, T. & Völter, M. (2006) *Model-Driven Software Development*: John Wiley & Sons,Inc. England.

VIII. Mapa curricular