

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

**Guía pedagógica:**  
**Calidad del software**

Elaboró: L.S.C.A. 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	11
VIII. Mapa curricular	12



### I. Datos de identificación

Espacio educativo donde se imparte	<b>Unidad Académica Profesional Tlanguistenco</b>			
Licenciatura	<b>Licenciatura en Ingeniería en Software</b>			
Unidad de aprendizaje	<b>Calidad del software</b>	Clave	<b>L40812</b>	
Carga académica	<b>2</b>	<b>1</b>	<b>3</b>	<b>5</b>
	Horas teóricas	Horas prácticas	Total de horas	Créditos
Período escolar en que se ubica	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
Seriación	Ninguna		Ninguna	
	UA Antecedente		UA Consecuente	

### Tipo de Unidad de Aprendizaje

Curso	<input checked="" type="checkbox"/>	Curso taller	<input type="checkbox"/>
Seminario	<input type="checkbox"/>	Taller	<input type="checkbox"/>
Laboratorio	<input type="checkbox"/>	Práctica profesional	<input type="checkbox"/>
Otro tipo (especificar)	<input type="text"/>		

### Modalidad educativa

Escolarizada. Sistema rígido	<input type="checkbox"/>	No escolarizada. Sistema virtual	<input type="checkbox"/>
Escolarizada. Sistema flexible	<input checked="" type="checkbox"/>	No escolarizada. Sistema a distancia	<input type="checkbox"/>
No escolarizada. Sistema abierto	<input type="checkbox"/>	Mixta (especificar)	<input type="text"/>

### Formación común

<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

### Formación equivalente

<b>Unidad de Aprendizaje</b>
<input type="text"/>
<input type="text"/>
<input type="text"/>



## 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 Calidad del 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 parámetros de calidad aplicables al desarrollo de software así como establecer criterios para aplicar el patrón de diseño y evaluación 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 Programación e ingeniería de software, núcleo que promueve una relación inter y trans-disciplinaria al agrupar unidades de aprendizaje que definen fundamentalmente la formación práctica o aplicada 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 las técnicas de calidad en el desarrollo, así como la misma Ingeniería de software.



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

<b>Núcleo de formación:</b>	Integral
<b>Área Curricular:</b>	Programación e ingeniería de software
<b>Carácter de la UA:</b>	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:

Programación e ingeniería de software:

Comprender las diferentes filosofías, conceptos, metodologías y técnicas utilizadas para la construcción de sistemas de software, considerando sus requerimientos, análisis y modelado, diseño, validación, verificación y calidad.

Analizar los diferentes elementos que inciden en la creación de productos de software desde una perspectiva de desarrollo industrial, incluyendo aspectos de eficiencia del proceso de creación, uso de herramientas automatizadas para su desarrollo, robustez, adaptabilidad, análisis de costos y tiempos y comercialización, entre otros.



## V. Objetivos de la unidad de aprendizaje.

Identificar normas y estándares para el aseguramiento de calidad en el desarrollo de software tanto en su proceso como en su producto, así como los métodos, técnicas y herramientas para la validación y verificación de software.

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

<b>Unidad 1.</b> Necesidad de disponer de técnicas para el desarrollo de software de calidad.		
<b>Objetivo:</b> Identificar la necesidad de crear, mejorar o actualizar software con técnicas de calidad. Diferenciar, argumentar y evaluar la contribución de la Ingeniería de software así como de las principales disciplinas que permiten identificar y aplicar los modelos de calidad existentes en el desarrollo de software.		
<b>Contenidos:</b> <b>1.1 Enfoque actual de la calidad de software.</b> <b>1.2 Importancia de aplicar principios para obtener software de calidad.</b> <b>1.3 Relevancia del control de la calidad del software.</b>		
<b>Métodos, estrategias y recursos educativos</b>		
Discusión y análisis, lluvia de ideas, trabajo en equipos, mapa conceptual, cuadro sinóptico, método de caso, ABP, síntesis, exposición y clase magistral.		
<b>Actividades de enseñanza y de aprendizaje</b>		
<b>Inicio</b>	<b>Desarrollo</b>	<b>Cierre</b>
1.1. Lluvia de ideas e investigación personal. Elabora un diagrama con las principales razones por las que hoy el software y su desarrollo son todavía deficientes.	1.1. Clase magistral. Anotación de los puntos más relevantes expuestos de la necesidad de la industria del software de contar con estándares de calidad para el desarrollo. Entrega síntesis.	1.1. En equipos de trabajo elabora mapa conceptual del enfoque actual de la calidad para cubrir la necesidad de la industria de disponer de estándares aplicables al desarrollo de software.
1.2. Método de caso. Discusión en grupos pequeños sobre los principios de calidad que se han adecuado y apoyado a la industria del software. Entrega reporte.	1.2. Debate donde se analizan y proponen diferentes puntos de vista sobre los principios de calidad que apoyan a la industria para el desarrollo y soporte. Entrega conclusiones.	1.2. Elabora cuadro sinóptico sobre cómo aportan a la creación, actualización y soporte de software los principios de calidad.



1.3. Clase magistral. Anotación de los puntos más relevantes expuestos de la relevancia del control de calidad en la ingeniería de software. Entrega resumen.	1.3. Aprendizaje basado en problemas. Mediante investigación y esquemas el alumno identificará el papel y la importancia del control de la calidad para desarrollar software adecuadamente.	1.3. En equipos de trabajo. Exposición y reporte donde se valora el papel que desempeñan hoy los principios y estándares de calidad en la adecuada aplicación de la Ingeniería de software.
<b>(1 Hrs.)</b>	<b>(4 Hrs.)</b>	<b>(1 Hrs.)</b>
<b>Escenarios y recursos para el aprendizaje (uso del alumno)</b>		
<b>Escenarios</b>		<b>Recursos</b>
Aula		Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.

<b>Unidad 2.</b> Gestión de la calidad del software.		
<b>Objetivo:</b> Explicar la importancia de la pertinente evaluación del análisis y el diseño de un sistema, así como la forma adecuada de medir y cuantificar la calidad en el todo proceso, con el fin de determinar su impacto en el producto final. Diferenciar las técnicas, mediciones y estimaciones de calidad que se aplican tanto en las etapas del desarrollo como para el producto final.		
<b>Contenidos:</b> <b>2.1 Aseguramiento de calidad.</b> <b>2.2 Técnicas, principios y estándares.</b> <b>2.3 Métricas y control de calidad.</b> <b>2.4 Pruebas, verificación y validación.</b>		
<b>Métodos, estrategias y recursos educativos</b>		
Discusión y análisis, investigaciones, trabajo en equipos, cuadro comparativo, exposición y clase magistral.		
<b>Actividades de enseñanza y de aprendizaje</b>		
<b>Inicio</b>	<b>Desarrollo</b>	<b>Cierre</b>
2.1. y 2.2. Clase magistral. Anotación de los puntos más relevantes expuestos de las técnicas y principios del aseguramiento de calidad aplicado en el desarrollo de software. Presenta resumen.	2.1. y 2.2. Investigación de los principios y estándares derivados del aseguramiento de calidad y entrega un cuadro sinóptico con los resultados.	2.1. y 2.2. En equipos de trabajo. Exposición y reporte donde presentan los beneficios y ventajas de implementar técnicas y estándares de calidad en las diferentes etapas del desarrollo del software.



2.3. y 2.4. Clase magistral. Anotación de los puntos más relevantes expuestos de lo que es control de calidad, así como la manera de implementarlo en el desarrollo de software. Entrega resumen.	2.3. y 2.4. Investigación y análisis aportando diferentes puntos de vista sobre las diversas métricas de control de calidad que se pueden aplicar en el desarrollo de software. Presenta esquema con las principales características.	2.3. y 2.4. Mediante un cuadro comparativo el alumno conoce e identifica las diferentes técnicas para probar, verificar y validar la calidad en un desarrollo de software así como su respectiva documentación.
<b>(2 Hrs.)</b>	<b>(8 Hrs.)</b>	<b>(3 Hrs.)</b>
<b>Escenarios y recursos para el aprendizaje (uso del alumno)</b>		
<b>Escenarios</b>		<b>Recursos</b>
Aula Laboratorio		Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.

<b>Unidad 3.</b> Modelos de calidad de software.		
<b>Objetivo:</b> Distinguir los rasgos particulares de cada tipo de modelos de calidad para el desarrollo de aplicaciones, identificando las ventajas e inconvenientes de cada uno de ellos, evaluando las condiciones y requisitos establecidos por la industria del software para proponer y aplicar los más adecuados.		
<b>Contenidos:</b> <b>3.1 Modelos por niveles.</b> <b>3.2 Modelo de madurez (CMMI).</b> <b>3.3 Modelos de procesos (PSP/TSP).</b> <b>3.4 Normas ISO.</b>		
<b>Métodos, estrategias y recursos educativos</b>		
Discusión y análisis, investigaciones, trabajo en equipos, cuadro comparativo, exposición y clase magistral.		
<b>Actividades de enseñanza y de aprendizaje</b>		
<b>Inicio</b>	<b>Desarrollo</b>	<b>Cierre</b>
3.1., 3.2. y 3.3. Clase magistral. Anotación de los puntos más relevantes expuestos de los tipos y características de los diferentes modelos de calidad de software. Entrega resumen.	3.1., 3.2. y 3.3. Investigación y análisis aportando diferentes puntos de vista sobre los principios y características de cada uno de los modelos. Presenta esquema con los principales atributos.	3.1., 3.2. y 3.3. Mediante un cuadro comparativo el alumno conoce e identifica los diferentes modelos de calidad de software aplicables al desarrollo así como sus adecuaciones en la práctica.



3.4. Clase magistral. Anotación de los puntos más relevantes expuestos de lo que son las normas ISO que aplican al desarrollo de software. Presenta resumen.	3.4. Investigación de los principales elementos de las normas ISO orientadas al desarrollo de software y entrega un cuadro sinóptico con los resultados.	3.4. En equipos de trabajo. Exposición y reporte donde presentan el alcance, los recursos y usos de aplicar las normas ISO en el desarrollo de software.
<b>(2 Hrs.)</b>	<b>(7 Hrs.)</b>	<b>(2 Hrs.)</b>
<b>Escenarios y recursos para el aprendizaje (uso del alumno)</b>		
<b>Escenarios</b>		<b>Recursos</b>
Aula Laboratorio		Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.

<b>Unidad 4.</b> Selección e implementación de técnicas y modelos para un caso particular de desarrollo.		
<b>Objetivo:</b> Determinar los requerimientos de administración, programación, soporte y costo de un desarrollo de software en particular, con el fin de aplicar las técnicas, estándares y criterios más convenientes así como implementar el modelo de calidad más pertinente. Y de acuerdo a sus características propias, proponer el plan de calidad más adecuado.		
<b>Contenidos:</b> <b>4.1 Definición de la gestión de calidad a aplicar.</b> <b>4.2 Técnicas y criterios para implementar un modelo.</b> <b>4.3 Selección e implementación de técnicas para desarrollar software de calidad específico.</b>		
<b>Métodos, estrategias y recursos educativos</b>		
Taller, investigaciones, cuadro sinóptico, resolución de problemas y clase magistral.		
<b>Actividades de enseñanza y de aprendizaje</b>		
<b>Inicio</b>	<b>Desarrollo</b>	<b>Cierre</b>
4.1., y 4.2. En equipos de trabajo investiga las técnicas y criterios que se emplean por la industria para gestionar calidad e implementar un modelo de para el ciclo de producción de un software específico. Entrega conclusiones.	4.1., y 4.2. Clase magistral. Anotación de los puntos más relevantes expuestos de lo necesario para gestionar calidad en casos específicos así como para implementar un modelo, de acuerdo a sus características propias. Entrega resumen.	4.1., y 4.2. Elabora cuadro sinóptico con los principales técnicas y criterios que ayudan en la gestión e implementación de los modelos de calidad a casos de desarrollo con características específicas.



4.3. En equipos de trabajo seleccionan y proponen un caso y el software para el que definirán el plan de de calidad a seguir en cierto desarrollo. Entrega propuesta de desarrollo.	4.3. Taller supervisado para resolver el problema planteado en equipos de trabajo de acuerdo a los criterios indicados. Entrega avances.	4.3. Exposición y entrega de trabajo final con la propuesta de calidad completa, correspondiente al desarrollo del software específico.
<b>(3 Hrs.)</b>	<b>(10 Hrs.)</b>	<b>(5 Hrs.)</b>
<b>Escenarios y recursos para el aprendizaje (uso del alumno)</b>		
<b>Escenarios</b>	<b>Recursos</b>	
Aula Laboratorio	Pintarrón, textos de apoyo, internet, computadora, cañón y diapositivas.	



## VII. Acervo bibliográfico

### Básico:

1. Piattini, M.; García, F. & Caballero, I. (2007), *Calidad de Sistemas Informáticos (1a edición)*: Alfaomega/RA-MA editorial, Madrid, España.
2. Kan, S. (2002), *Metrics and Models in software Quality Engineering (2nd edition)*: Pearson education, USA.
3. Jackson, P. & Ashton, D. (1996), *Implemente calidad de clase mundial. ISO 9000-BS5750*: Limusa. México.
4. Sanders, J. & Curran, E. (1994), *Software Quality*: Addison-Wesley.

### Complementario:

1. Watts, H. (1989), *Managing the Software Process*: Addison Wesley.
2. Gilb, T., Graham, D. (1993), *Software Inspection*: Addison Wesley.
3. Nistal Rosique, G. (1999), *Presentación del monográfico de Calidad del Software*: Novática, 137(En-Feb).
4. Pressman, R. (2006), *Ingeniería del Software: Un Enfoque Practico*: McGraw-Hill.
5. Myers, G. J. (1979), *The Art of Software Testing*: Wiley-Interscience.
6. Osteen, O., Glass, R. L. (1996), *An ISO 9000 Approach to building Quality Software*: Prentice Hall.



### VIII. Mapa curricular

