

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

**Guía de evaluación del aprendizaje:**

**Herramientas para la Administración  
y Programación de Sistemas**

Elaboró: L.I. Leonor Gonzalez Muñoz Fecha: 31/agosto/2015  
L.S.C.A. Carlos Alberto García Acevedo

Fecha de  
aprobación

H. Consejo académico

H. Consejo de Gobierno

\_\_\_\_\_

\_\_\_\_\_



## Índice

	Pág.
I. Datos de identificación	3
II. Presentación de la guía de evaluación del aprendizaje	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 actividades de evaluación	6
VII. Mapa curricular	12



### I. Datos de identificación

Espacio educativo donde se imparte **Unidad Académica Profesional Tianguistenco**

Licenciatura **Licenciatura en Ingeniería de Software**

Unidad de aprendizaje **Herramientas para la administración y programación de sistemas** Clave **L40815**

Carga académica

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</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>	<b>9</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------

Seriación

Sistemas Operativos	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

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

### Formación equivalente

**Unidad de Aprendizaje**

<input type="text"/>
<input type="text"/>
<input type="text"/>



## II. Presentación de la guía de evaluación.

Esta guía pedagógica de la unidad de aprendizaje de herramientas de la administración y programación de sistemas pretende conocer e identificar características que permitan establecer criterios para aplicar las técnicas más adecuadas a cada caso de construcción, mejora o actualización, fundamentándose en las tecnologías existentes.

El programa plantea la necesidad de aprovechar las aportaciones de la Ingeniería de software, así como las principales disciplinas de administración, programación y diseño. Además de la identificación de diferentes técnicas de evaluación de sistemas críticos, que permiten hacer software bajo las condiciones que prevalecen en las organizaciones de hoy.

Este núcleo promueve una relación inter y transdisciplinaria al agrupar unidades de aprendizaje que definen fundamentalmente la formación práctica o aplicada de la profesión y el desarrollo de competencias específicas. Para conocer conceptos, fundamentos de Sistemas operativos y Teoría de Lenguajes de Programación. Así como aplicar, distinguir y proponer adecuadamente las técnicas para definir los requisitos enfocados a solucionar problemas prácticos.

También provee 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 ampliando su visión al percibir cómo la industria está actualmente utilizando diversas técnicas para desarrollar software.

Mostrando la evaluación pertinente en cada una de las unidades que conforman el programa educativo de que consta esta guía de evaluación permitirá evaluar la habilidad de emplearlos para crear, mejorar o actualizar software proporcionando dentro de los parámetros de evaluación una actividad y el instrumento dentro de la prueba ordinaria, extraordinaria y título de suficiencia que dan a la unidad de aprendizaje de herramientas para la administración y programación de software la posibilidad de preparar a un profesionista capaz de dar pronta solución a los problemas de actualidad en la ingeniería de software.

..



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

<b>Núcleo de formación:</b>	Integral
<b>Área Curricular:</b>	Software de Base
<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:

Esta unidad de aprendizaje en conjunto con las otras unidades que conforman el núcleo integral de formación de la Licenciatura en Ingeniería de software que tiene como propósito:

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:

El área curricular establece 3 objetivos los cuales están relacionados con esta unidad de aprendizaje y a continuación se indican.



Aplicar la teoría, técnicas y metodologías para el diseño y construcción de intérpretes y compiladores, presentando las principales herramientas para su generación automática.

Aplicar la teoría, técnicas y metodologías para el diseño y construcción de sistemas operativos, aplicaciones para microcontroladores de propósito específico, con énfasis en cada uno de sus componentes: manejo del procesador, manejo de memoria, administración de dispositivos y manejo de información.

Emplear las principales categorías y herramientas de base necesarias para la configuración, el arranque, el uso eficiente y la operación de los sistemas de cómputo.

#### **V. Objetivos de la unidad de aprendizaje.**

Aplicar las diferentes herramientas de la administración y programación de sistemas, mediante el uso de teorías relacionadas con la administración y desarrollo de sistemas para la construcción de software especializado.

#### **VI. Contenidos de la unidad de aprendizaje, y actividades de evaluación.**

<b>Unidad 1. Introducción</b>
<b>Objetivo:</b> Distinguir herramientas de sistemas, identificando los requerimientos específicos y los rasgos que los categorizan para crear, mejorar o actualizar software en base a los requerimientos propios de cada caso, evaluando la contribución de las herramientas de sistemas para el desarrollo de programas.
<b>Contenidos:</b> 1.1 Razones para emplear herramientas para desarrollar sistemas



1.2 El contexto de las herramientas de administración y programación de sistemas: inicio, presente y futuro		
1.3 Uso de herramientas para desarrollar software especializado		
<b>Evaluación del aprendizaje</b>		
<b>Actividad</b>	<b>Evidencia</b>	<b>Instrumento</b>
1.3 Clase magistral.	Cuadro sinóptico, mapa mental	Rubrica de evaluación
1.2 Método Cognitivo	Cuadro comparativo, reporte de ejercicios	Rubrica de evaluación
1.3Tecnica interactiva	Cuadro sinóptico Reporte	Rubrica de evaluación

<b>Unidad 2.</b> Herramientas de administración de sistemas.		
<b>Objetivo:</b> Diferenciar las características propias de cada herramienta de administración, evaluando las ventajas y desventajas para mejorar y aplicar la herramienta más conveniente en diferentes casos según las necesidades de desarrollo, y la efectividad de los procesos de programación de sistemas.		
<b>Contenidos:</b>  2.1 Análisis de sistemas 2.2 Herramientas CASE 2.3 Administración de riesgos 2.4 Diseño de sistemas		
<b>Evaluación del aprendizaje</b>		
<b>Actividad</b>	<b>Evidencia</b>	<b>Instrumento</b>
2.1 Método de caso	Informe	Lista de cotejo



2.2 Técnicas de corrillos	Memorama	Rubrica Analitica
2.3 Lluvia de ideas Debate	Informe y Diagrama	Lista de cotejo
2.4 Técnicas de investigación	Reporte	Lista de cotejo

<b>Unidad 3. Herramientas de programación de sistemas</b>		
<b>Objetivo:</b>		
Distinguir los rasgos particulares de cada herramienta de programación identificando las ventajas e inconvenientes de cada herramienta de programación para evaluar las condiciones y requisitos establecidos por la industria del software proponiendo y aplicando las herramientas de programación más adecuadas.		
<b>Contenidos:</b>		
3.1 Teoría de Programación de sistemas 3.2 Administración de lenguajes de programación 3.3 Administración de sistemas operativos y redes 3.4 Administración de centros de cómputo		
<b>Evaluación del aprendizaje</b>		
<b>Actividad</b>	<b>Evidencia</b>	<b>Instrumento</b>
3.1 Técnicas de selección	Mapa mental	Rubrica
3.2 Técnicas de corrillos	Reporte	Lista de Cotejo
3.3 Resumen	Resumen escrito	Rubrica analítica
3.4 Clase Magistral	Diagrama Cuadro sinóptico	Listo de Cotejo





**Unidad 4.** Selección e implementación de las herramientas adecuadas para un caso específico de desarrollo.

**Objetivo:**

Determinar los requerimientos de administración, programación, soporte y costo del desarrollo a partir de la estructura y el análisis de una organización aplicando las técnicas, estándares y criterios más convenientes para diseñar un programa de administración y programación en un caso particular.

**Contenidos**

- 4.1 Selección de las herramientas de administración más adecuadas
- 4.2 Técnicas y criterios para implementar herramientas de programación
- 4.3 Diseño e implementación herramientas de sistemas para desarrollar software específico.

**Evaluación del aprendizaje**

Actividad	Evidencia	Instrumento
4.1,4.2 Técnicas de preguntas recíprocas	Diagrama de solución	Lista de cotejo
4.3. Clase Magistral	Proyecto	Rubrica

**Primera evaluación parcial**

Portafolio de evidencias	Rubrica	20%
Examen Escrito	Prueba pedagógica	30%
Proyecto de investigación	Rubrica	20%
Apuntes	Rubrica	15%
Tareas	Rubrica	15%
		100 %



### Segunda evaluación parcial

Portafolio de evidencias	Rubrica	20%
Examen Escrito	Prueba pedagógica	30%
Proyecto de investigación	Rubrica	20%
Apuntes	Rubrica	15%
Tareas	Rubrica	15%
		100 %

### Evaluación ordinaria final

Evidencia	Instrumento	Porcentaje
Examen escrito	Prueba pedagógica	100%

### Evaluación extraordinaria

Evidencia	Instrumento	Porcentaje
Examen escrito	Prueba pedagógica	100%

### Evaluación a título de suficiencia

Evidencia	Instrumento	Porcentaje
Examen escrito	Prueba pedagógica	100%



## VII. Acervo bibliográfico

### Básica:

1. Greenfield, J., Short, K., Cook, S., & Kent, S. (2003) *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. New York, NY, USA: ACM, ISBN: 1581137516
2. Jacky, J. (2001) *The Way of Z: Practical Programming with Formal Methods*. USA: Cambridge University Press.
3. Pressman R. S. (2006), *Ingeniería de Software: Un enfoque Práctico*, México: McGraw-Hill, ISBN: 8448100263

### Complementaria:

1. Jacobson, I., Booch, G., & Rumbaugh, J. (2000) *El Proceso Unificado de Desarrollo de Software*. Madrid: Adison-Wesley
2. Kelly, S. & Tolvanen, J. (2008) *Domain-Specific Modeling: Enabling Full Code Generation*. New Jersey, USA: John Wiley & Sons, Inc.
3. Pastor, O., Molina, J. (2004) *Model driven architecture in practice*. Valencia, España: Springer.
4. Stahl, T., Völter, M. (2006) *Model-Driven Software Development*. England: John Wiley & Sons, Inc.



Licenciatura en Ingeniería de Software

VIII. Mapa curricular

