

## Programa descriptivo por unidad de competencia

<b>Programa educativo</b>	<b>Licenciatura en Ingeniería en Desarrollo y Tecnologías de Software</b>	<b>Modalidad</b>		Presencial	
<b>Clave</b>	TD02	<b>H S M</b>		<b>Horas semestrales</b>	<b>Créditos</b>
<b>Unidad de competencia</b>	<b>Taller de desarrollo 2</b>	<b>Teoría</b>	<b>Práctica</b>	64	5
		1	3		
<b>Ubicación</b>	Cuarto semestre.	<b>Unidades CONAIC</b>		32.00	
<b>Prerrequisito</b>	Taller de desarrollo 1.	<b>H S M de cómputo</b>		3	
<b>Perfil docente</b>	Contar con título profesional, grado de maestría y preferentemente con grado de doctorado en áreas afines a informática y computación. Demostrar experiencia en docencia en el nivel medio superior o superior mínima de dos años. Dominar los lenguajes de programación actuales, deseable con una certificación en lenguajes de programación.				
<b>Presentación</b>	En esta Unidad de Competencia (UC) el estudiante tiene la capacidad para identificar los problemas presentados en el desarrollo de software e implementar el tipo de patrón de diseño que mejor se adapta a la solución para asegurar la calidad en el desarrollo de productos de software. Se ubica en el área de conocimiento de programación e ingeniería de software, y se relaciona con la UC de Programación Orientada a Objetos y Taller 1. Atiende al perfil de egreso en el sentido que domine los conocimientos teóricos y prácticos para la construcción de soluciones de software robustas y escalables, como las habilidades de programación, identificación de patrones de diseño de software .				
<b>Propósito</b>	Desarrolla software mediante la implementación de patrones de diseño de software con la finalidad de crear productos de software robustos y escalables.				
<b>Competencias genéricas</b>					
Aplica un pensamiento sistémico y complejo en la construcción de conocimientos y toma de decisiones.					
<b>Competencias disciplinares</b>					
Posee los conocimientos teóricos y prácticos para la construcción conceptual de soluciones de software.					
<b>Competencias profesionales</b>					
Desarrolla productos de software aplicando modelos y metodologías para el aseguramiento de la calidad en los procesos de desarrollo.					

## Mapa de la unidad de competencia

Unidad de competencia	Subcompetencia	Resultado de aprendizaje
<p style="text-align: center;"><b>Taller de desarrollo 2</b></p>	<p>1. Implementa los conceptos avanzados de programación orientada a objetos.</p>	<p>1.1. Domina los distintos conceptos avanzados de la programación para su implementación.</p>
	<p>2. Aplica los tipos de patrones de diseño e implementa patrones creacionales, estructurales y de comportamiento.</p>	<p>2.1. Aprende las diferencias existentes entre los distintos tipos de patrones y los problemas que estos resuelven. 2.2. Implementa Patrones de diseño creacionales, estructurales y de comportamiento en soluciones de software.</p>
	<p>3. Comprende las distintas arquitecturas de software.</p>	<p>3.1. Implementa arquitecturas en el desarrollo de una solución de software.</p>

## Cuadro descriptivo por subcompetencia

<b>Subcompetencia</b>	<b>Implementa los conceptos avanzados de Programación Orientada a Objetos.</b>	<b>Número</b>	<b>1</b>
<b>Propósito de la subcompetencia</b>	Entiende los conceptos avanzados de la programación orientada a objetos y los aplica para la creación de módulos para sistemas o componentes generales.	<b>Total de horas</b>	12
<b>Resultado de aprendizaje</b>	1.1. Domina los distintos conceptos avanzados de la programación para su implementación.	<b>Horas asignadas</b>	12
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>
<ol style="list-style-type: none"> <li>Investigación de la programación orientada a objetos.</li> <li>Investigación del uso de clases abstractas e interfaces.</li> <li>Creación de aplicaciones modulares.</li> <li>Evaluación del resultado de aprendizaje.</li> </ol>	<ol style="list-style-type: none"> <li>Documento Digital de la investigación de la POO.</li> <li>Modelado de aplicaciones con UML 2.0.</li> <li>Documento Digital de la investigación del uso de clases abstractas e interfaces.</li> <li>Código fuente de programas que implementan clases abstractas e interfaces.</li> <li>Código fuente de programas con integración entre módulos independientes que sean integrados en un sistema principal.</li> <li>Examen escrito.</li> </ol>	20%	<ol style="list-style-type: none"> <li>Modelado UML 2.0 y POO.</li> <li>Sobrecarga, herencia y polimorfismo.</li> <li>Interfaces y clases abstractas.</li> <li>Módulos.</li> </ol>

## Cuadro descriptivo por subcompetencia

<b>Subcompetencia</b>	<b>Aplica los tipos de patrones de diseño e implementa patrones creacionales, estructurales y de comportamiento.</b>	<b>Número</b>	<b>2</b>
<b>Propósito de la subcompetencia</b>	Entiende e identifica problemas que se resuelven con la implementación de patrones de diseño para la construcción de soluciones de software robustas y escalables.	<b>Total de horas</b>	30
<b>Resultado de aprendizaje</b>	2.1. Aprende las diferencias existentes entre los distintos tipos de patrones y los problemas que estos resuelven.	<b>Horas asignadas</b>	6
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>
<ol style="list-style-type: none"> <li>Investiga los patrones de diseño de software.</li> <li>Investiga los problemas que resuelven los patrones creacionales, estructurales y de comportamiento.</li> <li>Evaluación del resultado de aprendizaje.</li> </ol>	<ol style="list-style-type: none"> <li>Documento digital acerca de los patrones de diseño.</li> <li>Documento digital de los tipos de patrones de diseño existentes y los problemas que resuelve su implementación.</li> <li>Examen escrito.</li> </ol>	20%	<ol style="list-style-type: none"> <li>Patrones de diseño.</li> <li>Tipos de patrones de Diseño.</li> </ol>

Resultado de aprendizaje	2.2. Implementa Patrones de diseño creacionales, estructurales y de comportamiento en soluciones de software.			Horas asignadas	24
Actividades de evaluación	Evidencias a recopilar	%	Contenido		
1. Programa aplicaciones implementando patrones creacionales. 2. Programa aplicaciones implementando patrones estructurales. 3. Programa aplicaciones implementando patrones de comportamiento. 4. Evaluación del resultado de aprendizaje.	1. Código fuente de la implementación de patrones creacionales en el desarrollo de soluciones de software. 2. Código fuente de la implementación de patrones estructurales en el desarrollo de soluciones de software. 3. Código fuente de la implementación de patrones de comportamiento en el desarrollo de soluciones de software. 4. Examen escrito.	30%	1. Patrones creacionales. 2. Patrones estructurales. 3. Patrones de comportamiento.		

## Cuadro descriptivo por subcompetencia

<b>Subcompetencia</b>	<b>Comprende las distintas arquitecturas de Software.</b>	<b>Número</b>	<b>3</b>
<b>Propósito de la subcompetencia</b>	Conoce las diferentes arquitecturas de software para implementarlas en el desarrollo de soluciones robustas y escalables.	<b>Total de horas</b>	22
<b>Resultado de aprendizaje</b>	3.1. Implementa arquitecturas en el desarrollo de una solución de software.	<b>Horas asignadas</b>	22
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>
<ol style="list-style-type: none"> <li>Investigación de los tipos de arquitectura de software.</li> <li>Definición de caso de estudio para el diseño de una arquitectura de software.</li> <li>Modelado de la arquitectura de software usando UML 2.0.</li> <li>Implementa la arquitectura de Software del caso de estudio.</li> <li>Evaluación del resultado de aprendizaje.</li> </ol>	<ol style="list-style-type: none"> <li>Documento digital de las distintas arquitecturas de software existentes.</li> <li>Documento impreso o digital de la arquitectura a emplear en un proyecto de software.</li> <li>Diseño con diagramas UML 2.0 de la arquitectura de software a emplear.</li> <li>Código fuente de la aplicación con los patrones de diseño empleados en el desarrollo del caso de estudio.</li> <li>Examen escrito.</li> </ol>	30%	<ol style="list-style-type: none"> <li>Tipos de arquitectura de software.</li> <li>Diseñar la arquitectura de software.</li> <li>Modelar la arquitectura con UML 2.0.</li> <li>Implementación de la arquitectura de software.</li> </ol>

<b>Actitudes y valores</b>	Liderazgo. Innovación. Cultura de trabajo.	
<b>Recursos, materiales y equipo didáctico</b>		
<b>Recursos didácticos</b>	<b>Equipo de apoyo didáctico</b>	
Apuntes. Diapositivas. Antologías. Guías de práctica.	Proyector. Laboratorio de cómputo. Computadoras. Software especializado.	
<b>Fuentes de información</b>		
<b>Bibliografía básica:</b> Pantaleo, G. (2014). <i>Ingeniería de software</i> . México: Alfaomega Grupo Editor. Braude, E. (2003). <i>Ingeniería de software una perspectiva orientada a objetos</i> . México: Alfaomega. Kendall y Kendall. (2011). <i>Análisis y diseño de sistemas</i> (8a. ed.). México: Pearson.		
<b>Bibliografía complementaria:</b> Sanchez, A.. (2012). <i>Ingeniería del Software. Un enfoque desde la Guía SWEBOK</i> . México: Alfaomega Grupo Editor.		
<b>Recursos digitales:</b> Ninguno.		