

Programa descriptivo por unidad de competencia

Programa educativo	Licenciatura en Ingeniería en Desarrollo y Tecnologías de Software	Modalidad		Presencial	
Clave	TD01	H S M		Horas semestrales	Créditos
Unidad de competencia	Taller de desarrollo 1	Teoría	Práctica	64	5
		1	3		
Ubicación	Tercer semestre.	Unidades CONAIC		32.00	
Prerrequisito	Estructura de datos y programación orientada a objetos.	H S M de cómputo		3	
Perfil docente	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.				
Presentación	En esta Unidad de Competencia (UC) el estudiante tiene la capacidad para entender y crear pruebas de software 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. Atiende al perfil de egreso en el sentido que domine los conocimientos teóricos y prácticos para la construcción de soluciones de software, como las habilidades de programación y del desarrollo de la unidad de competencia.				
Propósito	Desarrolla software y la especificación del requisito solicitado mediante pruebas, implementa pruebas y refactoriza para eliminar duplicidad en la programación y realizar mejoras al software.				
Competencias genéricas					
Aplica un pensamiento sistémico y complejo en la construcción de conocimientos y toma de decisiones.					
Competencias disciplinares					
Posee los conocimientos teóricos y prácticos para la construcción conceptual de soluciones de software.					
Competencias profesionales					
Identifica los elementos necesarios para la implementación de software.					

Mapa de la unidad de competencia

Unidad de competencia	Subcompetencia	Resultado de aprendizaje
Taller de desarrollo 1	1. Entiende los conceptos de desarrollo dirigido por pruebas.	1.1. Entiende los conceptos de agilismo, el algoritmo del desarrollo asistido por pruebas así como las buenas prácticas consideradas con este paradigma.
	2. Comprende los tipos de pruebas de software.	2.1. Aprende a realizar la definición y descripción de pruebas funcionales, de aceptación, unitarias e integrales de un proyecto de software.
	3. Implementa Pruebas Unitarias e Integrales.	3.1. Utiliza herramientas de software para la creación de pruebas unitarias e integrales de aplicaciones.
	4. Desarrollo Basado en Pruebas de Aceptación.	4.1. Crea pruebas unitarias y de integración a partir de la definición de historias de usuario.

Cuadro descriptivo por subcompetencia

Subcompetencia	Entiende los conceptos de desarrollo dirigido por pruebas.	Número	1
Propósito de la subcompetencia	Comprende los conceptos de agilismo y la necesidad de realizar un desarrollo asistido por pruebas en las aplicaciones.	Total de horas	12
Resultado de aprendizaje	1.1. Entiende los conceptos de agilismo, el algoritmo del desarrollo asistido por pruebas así como las buenas prácticas consideradas con este paradigma.	Horas asignadas	12
Actividades de evaluación	Evidencias a recopilar	%	Contenido
<ol style="list-style-type: none"> Realiza una investigación acerca del agilismo y la metodología XP. Expone el algoritmo del desarrollo asistido por pruebas. Investiga las buenas prácticas del desarrollo asistido por pruebas. Evaluación del resultado de aprendizaje. 	<ol style="list-style-type: none"> Documento de la investigación del agilismo y la metodología XP. Presentación Digital e interactiva de ciclo de vida del desarrollo asistido por pruebas. Documento digital de la investigación de las buenas prácticas del desarrollo asistido por pruebas. Examen escrito. 	25%	<ol style="list-style-type: none"> Agilismo. Algoritmo Desarrollo asistido por pruebas. Consideraciones y recomendaciones.

Cuadro descriptivo por subcompetencia

Subcompetencia	Comprende los tipos de pruebas de software.	Número	2
Propósito de la subcompetencia	Entiende las diferentes pruebas que se realizan en el desarrollo de aplicaciones.	Total de horas	16
Resultado de aprendizaje	2.1. Aprende a realizar la definición y descripción de pruebas funcionales, de aceptación, unitarias e integrales de un proyecto de software.	Horas asignadas	16
Actividades de evaluación	Evidencias a recopilar	%	Contenido
<ol style="list-style-type: none"> Realiza un investigación de las pruebas de aceptación y funcionales de una aplicación. Realiza un investigación de las pruebas de Unitarias y de Integración. Describe las pruebas de Aceptación y Funcionales de un proyecto de software. Describe las pruebas de Unitarias e Integrales de un proyecto de software. Evaluación del resultado de aprendizaje. 	<ol style="list-style-type: none"> Documento Digital de la investigación de pruebas Aceptación. Documento Digital de la investigación de pruebas Funcionales. Documento Digital de la investigación de pruebas Unitarias. Documento Digital de la investigación de pruebas de Integración. Documento de la descripción de pruebas Funcionales y de Aceptación. Documento con la descripción de las pruebas Unitarias e Integrales de un proyecto de software. Examen escrito. 	25%	<ol style="list-style-type: none"> Pruebas de Aceptación. Aceptación. Pruebas Unitarias. Pruebas de Integración.

Cuadro descriptivo por subcompetencia

Subcompetencia	Implementa pruebas unitarias e integrales.	Número	3
Propósito de la subcompetencia	Emplea el uso de herramientas para la automatización de pruebas unitarias y de integración partiendo de las pruebas funcionales y de aceptación.	Total de horas	18
Resultado de aprendizaje	3.1. Utiliza herramientas de software para la creación de pruebas unitarias e integrales de aplicaciones.	Horas asignadas	18
Actividades de evaluación	Evidencias a recopilar	%	Contenido
<ol style="list-style-type: none"> 1. Investiga las herramientas para la automatización de pruebas en el lenguaje a emplear. 2. Investiga los principios S.O.L.I.D. en el diseño orientado a Objetos para emplearse en pruebas automatizadas. 3. Investigación del uso de herramientas y generación de inyección de dependencias. 4. Evaluación del resultado de aprendizaje. 	<ol style="list-style-type: none"> 1. Reseña digital de las herramientas y funcionalidades a emplear en el taller. 2. Documento digital de los principios S.O.L.I.D. del Diseño Orientado a Objetos. 3. Código Fuente de Inyección de Dependencias para la automatización de pruebas. 4. Examen escrito. 	25%	<ol style="list-style-type: none"> 1. Herramientas para realización de Pruebas Unitarias. 2. Diseño Orientado a Objetos. 3. Principios S.O.L.I.D. 4. Inversión de Control.

Cuadro descriptivo por subcompetencia

Subcompetencia	Desarrollo basado en pruebas de aceptación.	Número	4
Propósito de la subcompetencia	Aprende a describir historias de usuario, obtener pruebas de las historias de usuario, crea pruebas unitarias e integrales e implementa pruebas de usuario.	Total de horas	18
Resultado de aprendizaje	4.1 Crea pruebas unitarias y de integración a partir de la definición de historias de usuario.	Horas asignadas	18
Actividades de evaluación	Evidencias a recopilar	%	Contenido
<ol style="list-style-type: none"> Define y describe las historias de usuario. Define pruebas a partir de historias de usuario. Crea y valida pruebas unitarias e integrales realizadas de manera automatizada. Evaluación del resultado de aprendizaje. 	<ol style="list-style-type: none"> Documento digital de la descripción de caso de estudio para obtener historias de usuario. Documento digital de preguntas realizadas para obtener pruebas a partir de las historias de usuario. Documento digital de las pruebas unitarias y de integración por cada historia de usuario. Examen escrito. 	25%	<ol style="list-style-type: none"> Historias de Usuario. Pruebas a realizar. Pruebas validadas y aceptadas. Contexto de pruebas.

Actitudes y valores	Honestidad. Responsabilidad. Innovación.	
Recursos, materiales y equipo didáctico		
	Recursos didácticos	Equipo de apoyo didáctico
	Apuntes. Diapositiva. Antologías. Manuales.	Proyector. Laboratorio de cómputo. Computadoras. Software especializado.
Fuentes de información		
Bibliografía básica: Aldama, V. (2010). <i>Administración por calidad</i> . México: Alfaomega. García, J. (2014). <i>Desarrollo de Software dirigido por modelos: Conceptos, métodos y herramientas</i> . México: Alfaomega.		
Bibliografía complementaria: Kendall y Kendall. (2011). <i>Análisis y diseño de sistemas</i> (8a. ed.). México: Pearson.		
Recursos digitales: Ninguno.		