

Programa descriptivo por unidad de competencia

Programa educativo	Licenciatura en Ingeniería en Desarrollo y Tecnologías de Software	Modalidad		Presencial	
Clave	IS01	H S M		Horas semestrales	Créditos
Unidad de competencia	Metodología de la programación	Teoría	Práctica	64	6
		2	2		
Ubicación	Primer semestre.	Unidades CONAIC		42.67	
Prerrequisito	Ninguno.	H S M de cómputo		2	
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 se desarrolla un pensamiento lógico para identificar el proceso de creación de un programa y el desarrollo de algoritmos para resolver problemas. Está diseñada para el logro de competencias específicas de los dominios: manejo de consola y diseño de algoritmos. Comprende los conceptos básicos de la programación y escribe expresiones aritméticas y lógicas en un lenguaje de programación, así como el uso y funcionamiento de las estructuras secuenciales, selectivas, arreglos unidimensionales y multidimensionales en el desarrollo de aplicaciones.				
Propósito	Analiza, diseña y desarrolla soluciones de problemas reales utilizando algoritmos computacionales para implementarlos en un lenguaje de programación.				
Competencias genéricas					
Aplica un pensamiento sistémico y complejo en la construcción de conocimientos y toma de decisiones. Maneja Tecnologías de la información y comunicación para la gestión y construcción de conocimientos. Se desempeña en su práctica profesional y vida personal bajo principios éticos y morales.					
Competencias disciplinares					
Aplica habilidades de abstracción y expresión matemática para la solución de problemas. Formula modelos matemáticos para la solución de problemas mediante el desarrollo de aplicaciones de software para diversos entornos.					
Competencias profesionales					
Ninguna.					

Mapa de la unidad de competencia

Unidad de competencia	Subcompetencia	Resultado de aprendizaje
<p>Metodología de la programación</p>	<p>1. Comprende los conceptos básicos de la programación.</p>	<p>1.1. Conoce la clasificación de software y sistemas. 1.2. Distingue qué es un algoritmo, un lenguaje de programación y los paradigmas existentes.</p>
	<p>2. Diseña algoritmos.</p>	<p>2. Identifica los datos de entrada y la salida de un algoritmo. 3. Diseña el algoritmo que solucione un problema. 4. Representa el algoritmo en un diagrama de flujo y pseudocódigo.</p>
	<p>3. Identifica los elementos de un programa.</p>	<p>3.1. Distingue la estructura de un programa. 3.2. Declara las variables y constantes que se utilizan en un algoritmo. 3.3. Codifica expresiones matemáticas.</p>
	<p>4. Comprende el uso de las estructuras de control.</p>	<p>4. 1. Construye expresiones booleanas para modelar situaciones reales. 4. 2. Diseña algoritmos para resolver problemas que impliquen la toma de decisiones, utilizando estructuras selectivas. 4. 3. Diseña algoritmos para resolver problemas que realicen una misma tarea varias veces usando estructuras repetitivas.</p>

	<p>5. Distingue las estructuras de datos.</p>	<p>5. 1. Determina las estructuras de datos involucradas en la solución de un problema.</p> <p>5. 2. Diseña soluciones empleando arreglos y estructuras (registros).</p> <p>5. 3. Utiliza arreglos y estructuras (registros) en programas escritos en algún lenguaje.</p>
--	---	---

Cuadro descriptivo por subcompetencia

Subcompetencia	Comprende los conceptos básicos de la programación.			Número	1
Propósito de la subcompetencia	Domina los conceptos básicos de la programación.			Total de horas	10
Resultado de aprendizaje	1.1. Conoce la clasificación de software y sistemas.			Horas asignadas	4
Actividades de evaluación	Evidencias a recopilar	%	Contenido		
1. Responde preguntas de los temas. 2. Analiza el tema.	1. Examen diagnóstico. 2. Mapas Conceptuales. 3. Distingue las estructuras de datos.	10%	1. Clasificación del software de sistemas y aplicación.		
Resultado de aprendizaje	1.2. Distingue qué es un algoritmo, un lenguaje de programación y los paradigmas existentes.			Horas asignadas	6
Actividades de evaluación	Evidencias a recopilar	%	Contenido		
1. Responde preguntas de los temas. 2. Analiza el tema.	1. Mapa conceptual. 2. Cuadro sinóptico. 3. Evaluación escrita.	10%	1. Algoritmo. 2. Lenguaje de Programación. 3. Programa. 4. Programación. 5. Paradigmas de programación.		

Cuadro descriptivo por subcompetencia

Subcompetencia	Diseña algoritmos.	Número	2
Propósito de la subcompetencia	Identifica los datos de entrada y la salida de un algoritmo que de solución a un problema, representándolo en un diagrama de flujo y pseudocódigo.	Total de horas	10
Resultado de aprendizaje	2.1. Identifica los datos de entrada y la salida de un algoritmo.	Horas asignadas	4
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Realiza ejercicios de los temas.	1. Mapas conceptuales. 2. Evaluación. 3. Ejercicios reales sobre algoritmos.	6%	1. Análisis de problemas. 2. Representación de algoritmos: gráfica y pseudocódigo.
Resultado de aprendizaje	2.2. Diseña el algoritmo que solucione un problema.	Horas asignadas	6
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Realiza ejercicios de los temas.	1. Cuadros sinópticos. 2. Mapas conceptuales. 3. Ejercicios reales sobre casos prácticos.	10%	1. Diseño de algoritmos aplicados a problemas. 2. Diseño algorítmico de funciones.

Cuadro descriptivo por subcompetencia

Subcompetencia	Identifica los elementos de un programa.	Número	3
Propósito de la subcompetencia	Distingue la estructura de un programa declarando variables, constantes para codificar expresiones matemáticas.	Total de horas	10
Resultado de aprendizaje	3.1. Distingue la estructura de un programa.	Horas asignadas	5
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Realiza ejercicios de los temas.	1. Mapas conceptuales. 2. Cuadro sinóptico. 3. Evaluación.	8%	1. Características del lenguaje de programación. 2. Estructura básica de un programa. 3. Traducción de un programa: compilación, enlace de un programa, errores en tiempo de compilación. 4. Ejecución de un programa.
Resultado de aprendizaje	3.2. Declara las variables y constantes que se utilizan en un algoritmo.	Horas asignadas	5
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Realiza ejercicios de los temas.	1. Mapas conceptuales. 2. Cuadros sinópticos. 3. Evaluación escrita. 4. Resolución de ejercicios.	8%	1. Elementos del lenguaje: datos, literales y constantes, identificadores, variables, parámetros, operadores, entrada y salida de datos. 2. Errores en tiempo de ejecución.

Cuadro descriptivo por subcompetencia

Subcompetencia	Comprende el uso de las estructuras de control.	Número	4
Propósito de la subcompetencia	Construye expresiones booleanas para modelar situaciones reales y diseña algoritmos para resolver problemas que impliquen la toma de decisiones, utilizando estructuras selectivas y para resolver problemas que realicen una misma tarea varias veces usando estructuras repetitivas.	Total de horas	19
Resultado de aprendizaje	4.1. Construye expresiones booleanas para modelar situaciones reales.	Horas asignadas	6
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Ejercicios de análisis.	1. Mapas conceptuales. 2. Cuadros sinópticos 3. Evaluación.	8%	1. Estructuras secuenciales. 2. Estructuras selectivas: simple, doble y múltiple.
Resultado de aprendizaje	4.2. Diseña algoritmos para resolver problemas que impliquen la toma de decisiones, utilizando estructuras selectivas.	Horas asignadas	7
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Realiza ejercicios de los temas.	1. Mapas conceptuales. 2. Cuadros sinópticos. 3. Evaluación. 4. Ejercicios prácticos.	8%	1. Diseño de algoritmos. 2. Estructuras selectivas.
Resultado de aprendizaje	4.3. Diseña algoritmos para resolver problemas que realicen una misma tarea varias veces usando estructuras repetitivas.	Horas asignadas	6
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Realiza ejercicios de los temas.	1. Mapas conceptuales. 2. Cuadros sinópticos. 3. Evaluación escrita. 4. Ejercicios prácticos.	8%	1. Estructuras iterativas: repetir mientras, hasta, y desde. 2. Diseño e implementación de funciones.

Cuadro descriptivo por subcompetencia

Subcompetencia	Distingue las estructuras de datos.	Número	5
Propósito de la subcompetencia	Determina las estructuras de datos involucradas en la solución de un problema y utiliza arreglos y estructuras (registros) en programas escritos en algún lenguaje de programación.	Total de horas	15
Resultado de aprendizaje	5.1. Determina las estructuras de datos involucradas en la solución de un problema.	Horas asignadas	5
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Realiza ejercicios de los temas.	1. Mapas conceptuales. 2. Cuadros sinópticos. 3. Examen. 4. Ejercicios prácticos.	8%	1. Arreglos. 2. Definición y tipos. 3. Declaración e inicialización.
Resultado de aprendizaje	5.2. Diseña soluciones empleando arreglos y estructuras (registros).	Horas asignadas	5
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Realiza ejercicios.	1. Mapas conceptuales. 2. Cuadros sinópticos. 3. Examen escrito. 4. Ejercicios prácticos.	8%	1. Acceso a los elementos de un arreglo. 2. Ciclos y arreglos. 3. Cadenas.
Resultado de aprendizaje	5.3. Utiliza arreglos y estructuras (registros) en programas escritos en algún lenguaje.	Horas asignadas	5
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Analiza el tema. 2. Responde preguntas de los temas. 3. Realiza ejercicios.	1. Mapa conceptual. 2. Cuadro sinóptico. 3. Evaluación escrita. 4. Ejercicios prácticos.	8%	1. Estructuras. 2. Definición, declaración e inicialización. 3. Acceso a los elementos.

Actitudes y valores	Analítico. Ordenado. Coherente. Proactivo. Asertivo.	
Recursos, materiales y equipo didáctico		
	Recursos didácticos	Equipo de apoyo didáctico
	Antologías. Diapositivas. Videos.	Proyector de video. Software especializado.
Fuentes de información		
Bibliografía básica: Cairó, O. (2006). <i>Fundamentos de Programación</i> . México: Prentice Hall. Battistutti, O. (2006). <i>Metodología de la programación: algoritmos, diagramas</i> . México: Alfaomega Ra-Ma. Joyanes A. L. (2008). <i>Fundamentos de programación: Algoritmos y estructura</i> (4a. ed.). México: Mc. Graw Hill.		
Bibliografía complementaria: Kernighan B. y Ritchie D. (1995). <i>Lenguaje de Programación C</i> (2a. ed.). México: Pearson. López, G. (2009). <i>Análisis y diseño de algoritmos</i> . México: Alfaomega.		
Recursos digitales: Ninguno.		