

Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Marzo 22, 2022				
Carrera:	Ingeniería en Tecnologías de Software	Asignatura:	Introducción a la Programación		
Academia:	Informática y Computación Virtual /	Clave:	19SICTS0202		
Módulo formativo:	Informática y Computación	Seriación:	19SICTS0303 - Programación Estructurada		
Tipo de curso:	Modalidad mixta	Prerrequisito:	19SICTS0101 - Introducción a la Computación		
Semestre:	Segundo	Créditos:	5.63	Horas semestre:	90 horas
Teoría:	0 horas	Práctica:	0 horas	Trabajo indpt.:	5 horas
				Total x semana:	5 horas

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
OE1	Solucionará problemas con sólidas bases científicas y fundamentos tecnológicos que le permitirán comprender, analizar, diseñar, organizar, producir, operar y dar soluciones prácticas a problemas relacionados con las áreas de Organización de Sistemas Computacionales e Ingeniería en Software para el sector productivo y social, promoviendo los principios de ética, responsabilidad y trabajo colaborativo.	CD1. El egresado implementará las diferentes etapas del ciclo de vida del software contemplando la protección de datos y prevención dedesastres, salvaguardando con ética la seguridad de la información.	50 % Egresados trabajarán en cualquier proceso del desarrollo de software o áreas afines a los sistemas computacionales, promoviendo los principios de ética, responsabilidad y trabajo colaborativo.
OE2	Aportará soluciones innovadoras y sustentables en el área de la electrónica en el que establezca el análisis, diseño, implementación, selección de componentes de hardware de uso específico, el software asociado y su conectividad a través de redes de comunicación para el sector productivo y social.	CD2. El egresado implementará las diferentes técnicas de análisis y diseño de circuitos electrónicos que den una solución innovadorasustentable a problemas con el hardware.	20% Egresados trabajarán en cualquier proceso de creación y aplicación de hardware o áreas afines en el sector productivo y social.
OE3	Implementará soluciones innovadoras y sustentables con tecnologías de información que sean acordes a las necesidades, a las tecnologías disponibles y emergentes, para lograr un aprovechamiento óptimo de los recursos humanos y financieros en el sector productivo y social.	CD3. El egresado implementará las diferentes tecnologías emergentes en equipos multidisciplinarios que den una solución innovadora ysustentable a las necesidades que se presenten en el ámbito productivo y social.	20 % Egresados trabajarán en la aplicación de Tecnologías de la información o áreas afines en el sector productivo o social.



Atributos de egreso de plan de estudios		Criterios de desempeño	Componentes
AE1	Aplicar una experimentación adecuada con apoyo de metodologías y juicio ingenieril que permitan interpretar datos para obtener conclusiones que den solución a problemáticas en un contexto determinado.	<p>CD1. Conocerá y comprenderá las distintas metodologías para la solución de problemas.</p> <p>CD2. Conocerá la teoría de los algoritmos y su representación en diagramas de flujo.</p> <p>CD3. Conocerá la simbología requerida para la elaboración de un diagrama de flujo, así como las generalidades de los pseudocódigos y los criterios para su elaboración. en una semana</p> <p>CD4. Conocerá los principios de los lenguajes de programación, compiladores e intérpretes y entornos de desarrollo. Explicará con sus palabras mediante un cuestionario, las características de los lenguajes de programación, la diferencia entre compiladores e intérpretes y el concepto de entornos de desarrollo. con una presentación</p>	<p>1.1. Problema.</p> <p>1.1.1. Concepto de problema.</p> <p>1.1.2. Tipos de problemas.</p> <p>1.2. Metodologías de Solución de Problemas.</p> <p>1.2.1. Método de Pólya.</p> <p>1.2.2. Metodologías de solución de problemas.</p> <p>2.1. Algoritmia elemental.</p> <p>2.1.1. Definición de algoritmo en el contexto del desarrollo de software.</p> <p>2.1.2. Características de los algoritmos.</p> <p>2.1.3. Tipos de Algoritmos.</p> <p>2.1.4. Aplicación y uso de los algoritmos para la solución de problemas.</p> <p>2.1.5. Buenas prácticas de documentación de algoritmos.</p> <p>2.1.6. Software para la ejecución de algoritmos.</p> <p>2.2. Datos y estructuras de control.</p> <p>2.2.1. Tipos de datos.</p> <p>2.2.2. Identificadores, constantes y variables.</p> <p>2.2.3. Operadores y expresiones.</p> <p>2.2.4. Estructuras de entrada/salida de datos.</p> <p>2.2.5. Estructuras de control selectivas.</p> <p>2.2.6. Estructuras de control repetitivas.</p> <p>2.2.7. Anidamiento.</p> <p>2.3 Verificación de la eficacia de un algoritmo.</p> <p>2.3.1 Simulación de comportamiento.</p>



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			<ul style="list-style-type: none"> 3.1. Diagramas de flujo. 3.1.2. Tipos. 3.1.3. Simbología. 3.1.4. Ejercicios. 3.2. Pseudocódigo. 3.2.1. Concepto. 3.2.2. Palabras reservadas. 3.2.3. Ejercicios. 3.3. Herramientas de comprobación de algoritmos. 3.3.1. Corridas de escritorio. 4.1 Proceso Computacional y programas. 4.1.1 Concepto de programa. 4.1.2 Concepto de información. 4.2 Tipos de lenguajes de programación. 4.2.1 Lenguaje de bajo nivel. 4.2.2 Lenguaje de mediano nivel. 4.2.3 Lenguaje de alto nivel. 4.3 Traductores de Lenguaje. 4.3.1 Características de compiladores. 4.3.2 Características de intérpretes. 4.3.3 Proceso de compilación e interpretación. 4.3.4 Tipos de Código. 4.4 Entorno de desarrollo. 4.4.1 Entorno de desarrollo integrado. 4.4.2 Entorno de desarrollo externo. 4.4.3 Línea o comando. 1.1 Identificadores. 1.2 Tipos de datos. 1.3 Variables y constantes.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			1.4 Operadores y jerarquía. 1.5 Expresiones. 1.6 Palabras reservadas. 1.7 Comentarios. 1.8 Funciones de entrada (scanf y gets). 1.9 Funciones de salida (printf). 2.1 Contadores y acumuladores. 2.2 Estructuras de control. 2.2.1 Secuenciales. 2.2.2 Selectivas. 2.2.2.1 Simples. 2.2.2.2 Dobles. 2.2.2.3 Múltiples. 2.2.3 Repetitivas. 2.2.3.1 Desde. 2.2.3.2 Mientras. 2.2.3.3 Hacer mientras. 3.1. Arreglos. 3.1.1. Arreglos unidimensionales. 3.1.2. Arreglos bidimensionales. 3.2. Funciones. 3.2.1. Prototipo. 3.2.2. Sin parámetro. 3.2.3. Con parámetro. 3.2.4. Que regresa valor. 3.2.5 Que no regresa valor.

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Iniciar a los estudiantes en el ambiente de la programación, partiendo del análisis y resolución de problemas mediante algoritmos y diagramas de flujo.		
Atributos (competencia específica) de la asignatura		
Desarrollar productos de software que permitan almacenar, disponer y procesar información en diversas plataformas tecnológicas mediante el proceso o ciclo de vida de desarrollo de software, utilizando paradigmas que permitirán la construcción de productos y servicios innovadores de tecnologías de la información.		
Aportación a la competencia específica		Aportación a las competencias transversales
Saber	Saber hacer	Saber Ser
- Conocer los aspectos relacionados con la elaboración de algoritmos, pseudocódigo, diagramas de flujo y pruebas de escritorio necesarias para documentar la solución a un problema planteado.	- Realizar pseudocódigo y diagramas de flujo para representar un algoritmo, tanto de forma manual como utilizando software especializado en diseño de algoritmos y diagramas de flujo. - Aplicar metodologías de solución de problemas a la vida cotidiana. Utiliza la tecnología para apoyar su aprendizaje.	- Realiza actividades y exámenes, de manera organizada, con limpieza y ortografía impecable. - Participa activamente y siguiendo las normas de conducta, en los espacios de aprendizaje colaborativo. - Entrega puntualmente las actividades de aprendizaje atendiendo los criterios de evaluación. - Respeta derechos de autor, incluyendo la bibliografía en formato APA.
Producto integrador de la asignatura, considerando los avances por unidad		
Desarrollar el algoritmo de solución a distintos problemas planteados. Realizar su correspondiente algoritmo, diagrama de flujo, pseudocódigo y pruebas de escritorio de manera adecuada, utilizando el software especializado en la elaboración de diagramas de flujo.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Métodos de solución de problemas"

Número y nombre de la unidad: 1. Métodos de solución de problemas							
Tiempo y porcentaje para esta unidad:		Teoría:	1 hora	Práctica:	1 hora	Porcentaje del programa:	2.22%
Aprendizajes esperados: Conocer las distintas metodologías para la resolución de problemas cotidianos.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1.1. Problema. 1.1.1. Concepto de problema. 1.1.2. Tipos de problemas. 1.2. Metodologías de Solución de Problemas. 1.2.1. Método de Pólya. 1.2.2. Metodologías de solución de problemas.	Saber: - Conocer y comprender las distintas metodologías para la solución de problemas. Saber hacer: - Seleccionar y aplicar la metodología adecuada en la solución de problema. - Aplicar metodologías de solución de problemas a la vida cotidiana. - Utiliza la tecnología para apoyar su aprendizaje. Ser: - Realiza actividades y exámenes, de manera organizada, con limpieza y ortografía impecable. - Participa activamente y siguiendo las	- Rescate de conocimientos previos. - Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. - Introducción a la resolución de problemas mediante herramientas web interactivas.	Evaluación diagnóstica: - Identificar los conocimientos previos. Instrumentos de evaluación: Prueba escrita. Evaluación: Formativa: - Estrategia: Resolución de problemas / realización de ejercicios prácticos. Instrumentos de evaluación: - Rúbrica. Evaluación sumativa: - Examen escrito. Instrumentos de evaluación: - Cuestionario.	Planteamiento de un problema laboral o cotidiano en el que aplica la metodología adecuada, describiendo los pasos que conducen a la solución.			



Continuación: Tabla 4.1. Desglose específico de la unidad "Métodos de solución de problemas"

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	<p>normas de conducta, en los espacios de aprendizaje colaborativo.</p> <ul style="list-style-type: none">- Entrega puntualmente las actividades de aprendizaje atendiendo los criterios de evaluación.- Respeta derechos de autor, incluyendo la bibliografía en formato APA.			
Bibliografía				
<p>Farrell, J. (2013). Introducción a la programación lógica y diseño. Cengage Learning.</p> <p>Deitel, H. M.; Deitel, P. J. (2004). Cómo programar en C/C++ y Java. Pearson educación.</p>				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Algoritmia."

Número y nombre de la unidad: 2. Algoritmia.							
Tiempo y porcentaje para esta unidad:		Teoría:	1 hora	Práctica:	1 hora	Porcentaje del programa:	2.22%
Aprendizajes esperados:		Conocer los aspectos relevantes relacionados con el diseño y elaboración de algoritmos, así como el proceso formal para describirlos.			Identificar las distintas metodologías para aplicarlas en la resolución de problemas cotidianos.		
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
2.1. Algoritmia elemental. 2.1.1. Definición de algoritmo en el contexto del desarrollo de software. 2.1.2. Características de los algoritmos. 2.1.3. Tipos de Algoritmos. 2.1.4. Aplicación y uso de los algoritmos para la solución de problemas. 2.1.5. Buenas prácticas de documentación de algoritmos. 2.1.6. Software para la ejecución de algoritmos. 2.2. Datos y estructuras de control. 2.2.1. Tipos de datos. 2.2.2. Identificadores, constantes y variables. 2.2.3. Operadores y expresiones. 2.2.4. Estructuras de entrada/salida de datos.	Saber: - Conocer la teoría de los algoritmos y su representación en diagramas de flujo. Saber hacer: - Plantear la solución de un problema mediante algoritmos y diagramas de flujo. - Aplicar metodologías de solución de problemas a la vida cotidiana. - Utiliza la tecnología para apoyar su aprendizaje. Ser: - Realiza actividades y exámenes, de manera organizada, con limpieza y	- Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. - Introducción a la resolución de problemas mediante herramientas web interactivas.	Evaluación: Formativa: - Estrategia: Resolución de problemas / realización de ejercicios prácticos. Instrumentos de evaluación: - Rúbrica. Evaluación sumativa: - Examen escrito. Instrumentos de evaluación: - Cuestionario.	Trabajo donde se plantean distintas problemáticas con la finalidad de plantear un algoritmo que representará la solución a cada uno de ellos, con su respectiva corrida de escritorio.			



Continuación: Tabla 4.2. Desglose específico de la unidad "Algoritmia."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
2.2.5. Estructuras de control selectivas. 2.2.6. Estructuras de control repetitivas. 2.2.7. Anidamiento. 2.3 Herramientas de comprobación de algoritmos. 2.3.1 Corridas de escritorio.	ortografía impecable. - Participa activamente y siguiendo las normas de conducta, en los espacios de aprendizaje colaborativo. - Entrega puntualmente las actividades de aprendizaje atendiendo los criterios de evaluación. - Respeta derechos de autor, incluyendo la bibliografía en formato APA.			
Bibliografía				
Farrell, J. (2013). Introducción a la programación lógica y diseño. Cengage Learning. Deitel, H. M.; Deitel, P. J. (2004). Cómo programar en C/C++ y Java. Pearson educación. Corona, R. (1994). Taller de lógica. México: Amate.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Diagramas de flujo y pseudocódigos."

Número y nombre de la unidad: 3. Diagramas de flujo y pseudocódigos.							
Tiempo y porcentaje para esta unidad:		Teoría:	1 hora	Práctica:	1 hora	Porcentaje del programa:	2.22%
Aprendizajes esperados:		Conocer y aplicar la simbología de los diagramas de flujo para representar gráficamente cualquier algoritmo de solución, así como la forma de elaborar el pseudocódigo correspondiente a dicho algoritmo.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
3.1. Diagramas de flujo. 3.1.1. Concepto. 3.1.2. Tipos. 3.1.3. Simbología. 3.1.4. Ejercicios. 3.2. Pseudocódigo. 3.2.1. Concepto. 3.2.2. Palabras reservadas. 3.2.3. Ejercicios. 3.3. Herramientas de comprobación de algoritmos. 3.3.1. Corridas de escritorio.	Saber: - Conocer la simbología requerida para la elaboración de un diagrama de flujo, así como las generalidades de los pseudocódigos y los criterios para su elaboración. Saber hacer: - Representar mediante diagramas de flujo y pseudocódigos, la solución a distintas problemáticas. - Aplicar metodologías de solución de problemas a la vida cotidiana. - Utiliza la tecnología para apoyar su aprendizaje.	- Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. - Introducción a la resolución de problemas mediante herramientas web interactivas.	Evaluación: Formativa: - Estrategia: Resolución de problemas / realización de ejercicios prácticos. Instrumentos de evaluación: - Rúbrica. Evaluación sumativa: - Examen escrito. Instrumentos de evaluación: - Cuestionario.	Trabajo donde se plantean distintas problemáticas con la finalidad de plantear tanto su diagrama de flujo como su pseudocódigo, los cuales representarán la solución a cada uno de los problemas, utilizando el software especializado para su elaboración.			



Continuación: Tabla 4.3. Desglose específico de la unidad "Diagramas de flujo y pseudocódigos."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	<p>Ser:</p> <ul style="list-style-type: none"> - Realiza actividades y exámenes, de manera organizada, con limpieza y ortografía impecable. - Participa activamente y siguiendo las normas de conducta, en los espacios de aprendizaje colaborativo. - Entrega puntualmente las actividades de aprendizaje atendiendo los criterios de evaluación. - Respeta derechos de autor, incluyendo la bibliografía en formato APA. 			
Bibliografía				
<p>Farrell, J. (2013). Introducción a la programación lógica y diseño. Cengage Learning.</p> <p>Deitel, H. M.; Deitel, P. J. (2004). Cómo programar en C/C++ y Java. Pearson educación.</p> <p>Corona, R. (1994). Taller de lógica. México: Amate.</p>				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.4. Desglose específico de la unidad "Lenguajes de Programación."

Número y nombre de la unidad: 4. Lenguajes de Programación.							
Tiempo y porcentaje para esta unidad:		Teoría:	1 hora	Práctica:	1 hora	Porcentaje del programa:	2.22%
Aprendizajes esperados:		Conocer e identificar las generalidades de los diferentes tipos de lenguajes de programación, los principios de compiladores e intérpretes así como entornos de desarrollo que permitan la generación de soluciones a problemas informáticos.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
4.1 Proceso Computacional y programas. 4.1.1 Concepto de programa. 4.1.2 Concepto de información. 4.2 Tipos de lenguajes de programación. 4.2.1 Lenguaje de bajo nivel. 4.2.2 Lenguaje de mediano nivel. 4.2.3 Lenguaje de alto nivel. 4.3 Traductores de Lenguaje. 4.3.1 Características de compiladores. 4.3.2 Características de intérpretes. 4.3.3 Proceso de compilación e interpretación. 4.3.4 Tipos de Código. 4.4 Entorno de desarrollo.	Saber: - Conocer los principios de los lenguajes de programación, compiladores e intérpretes y entornos de desarrollo. - Explicar con sus palabras mediante un cuestionario, las características de los lenguajes de programación, la diferencia entre compiladores e intérpretes y el concepto de entornos de desarrollo. - Argumentar y debatir sobre las generalidades de los lenguajes de programación, tipos y características con la finalidad de enriquecer su	- Exposición por parte del profesor de material teórico. - Complementar información con material audiovisual. - Introducción a la resolución de problemas mediante herramientas web interactivas.	Evaluación: Formativa: - Estrategia: Resolución de problemas / realización de ejercicios prácticos. Instrumentos de evaluación: - Rúbrica. Evaluación sumativa: - Examen escrito. Instrumentos de evaluación: - Cuestionario.	Cuestionario donde se muestra su conocimiento sobre lenguajes de programación, tipos y características, diferencias entre compiladores e intérpretes, así como las fases para diseñar un programa informático.			



Continuación: Tabla 4.4. Desglose específico de la unidad "Lenguajes de Programación."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
4.4.1 Entorno de desarrollo integrado.	conocimiento.			
4.4.2 Entorno de desarrollo externo.				
4.4.3 Línea o comando.	<p>Saber hacer:</p> <ul style="list-style-type: none"> - Aplicar metodologías de solución de problemas a la vida cotidiana. - Utiliza la tecnología para apoyar su aprendizaje. <p>Ser:</p> <ul style="list-style-type: none"> - Realiza actividades y exámenes, de manera organizada, con limpieza y ortografía impecable. - Participa activamente y siguiendo las normas de conducta, en los espacios de aprendizaje colaborativo. - Entrega puntualmente las actividades de aprendizaje atendiendo los criterios de evaluación. - Respeta derechos de autor, incluyendo la bibliografía en formato APA. 			
Bibliografía				
<p>Farrell, J. (2013). Introducción a la programación lógica y diseño. Cengage Learning.</p> <p>Deitel, H. M.; Deitel, P. J. (2004). Cómo programar en C/C++ y Java. Pearson educación.</p> <p>Corona, R. (1994). Taller de lógica. México: Amate.</p>				



V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
<p>Carrera(s): - Ingeniería en Tecnologías de software.</p> <ul style="list-style-type: none">- Ingeniería en Informática, Computación, o carrera afín.- Ingeniería en Tecnología de la Información y Comunicación.- Licenciatura en informática.- Ingeniería en Tecnologías de la Información y Comunicación.- Ingeniería en Tecnologías de la Información y Comunicaciones.- Licenciatura en comunicación e información.- Ingeniería en sistemas de computación.- Ingeniería en sistemas computacionales.- Licenciatura en ingeniería en sistemas computacionales.- Licenciatura en ingeniería en sistemas de información.- Ingeniería en sistemas estratégicos de información, especialización en redes.- Licenciatura en sistemas computacionales.

- Ingeniería en sistemas de computación.

o carrera afín

Experiencia:

- Experiencia profesional comprobable correspondiente al campo disciplinar o carrera.
- Experiencia mínima de dos años
- Licenciatura en Informática o Sistemas Computacionales o Maestría relacionada con el área de conocimiento.