



Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Junio 01, 2022				
Carrera:	Ingeniería Mecatrónica	Asignatura:	Programación estructurada		
Academia:	Computación / Ciencias Sociales	Clave:	19SME03		
Módulo formativo:	Programación	Seriación:	19SME04 - Programación orientada a objetos		
Tipo de curso:	Presencial	Prerrequisito:	- -		
Semestre:	Primero	Créditos:	5.62	Horas semestre:	90 horas
Teoría:	2 horas	Práctica:	3 horas	Trabajo indpt.:	0 horas
				Total x semana:	5 horas

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
OE1	El egresado solucionará problemas del entorno laboral en el que se desempeñe, mediante el uso de conocimientos técnicos adquiridos para la identificación, desarrollo innovador, aplicación y control de las posibles soluciones, utilizando sus habilidades en mecánica, electrónica, control y automatización para dar el resultado adecuado según las condiciones del problema.	El egresado aplicará las técnicas y metodologías para la identificación de problemas referentes a su entorno laboral, proponiendo soluciones creativas e innovadoras para los mismos.	% de alumnos que implementan diversidad de técnicas y metodologías para identificar problemas en su entorno laboral.
OE2	El egresado diseñará, mejorará o mantendrá de forma eficiente y sustentable equipos que cubran adecuadamente las diferentes necesidades del ámbito laboral, utilizando sus competencias técnicas de diseño, con sus conocimientos de materiales, control y procesos para lograr la mejor solución innovadora de la necesidad planteada.	El egresado fundamentará documentalmente la solución a problemas, desde la identificación hasta su resolución.	% de egresados que diseñan, mejoran o dan mantenimiento a equipos.
OE3	El egresado generará relaciones interpersonales y profesionales de otras áreas, para desarrollar habilidades técnicas, administrativas y colaborativas en el desarrollo de proyectos mecatrónicos.	El egresado desarrollará canales de comunicación y de gestión con departamentos y áreas relacionadas con los proyectos que lidera y coordina.	% de egresados que participan en más de un departamento y/o área por proyecto con las que se relaciona.



Atributos de egreso de plan de estudios		Criterios de desempeño	Componentes
AE1	Identificar y resolver problemas en el campo de la mecatrónica aplicando los principios de las ciencias básicas como la matemáticas y física, así como otras ciencias de la ingeniería.	<ul style="list-style-type: none"> - Analizará escenarios hipotéticos identificando las entradas, procesos y salidas. - Propondrá un algoritmo que cumpla con las características de solución a un escenario hipotético. 	1. Diseño de un programa y algoritmia. 1.1 Análisis de un problema. 1.2 Diseño. 1.3 Implementación y pruebas. 1.4 Algoritmia. 1.4.1 Definición de un problema. 1.4.2 Análisis de modelo matemático. 1.4.3 Desarrollo del algoritmo. 1.4.4 Diagrama de flujo. 1.4.5 Pseudocódigo. 1.4.6 Diagrama de flujo de datos. 1.4.7 Código.
AE2	Desarrollar, innovar y/o implementar sistemas, procesos y productos para la solución integral de necesidades concretas en el campo de la mecatrónica.	<ul style="list-style-type: none"> - Desarrollará códigos en lenguaje C que solucione situaciones planteadas haciendo uso de estructuras de control, arreglo, funciones y registros. - Probará y evaluará los resultados obtenidos con el código propuesto para hacer el mantenimiento pertinente. 	2. Introducción al lenguaje C. 2.1 Intérpretes y compiladores. 2.2 Estructura de un programa. 2.3 Programación estructurada. 2.4 Historia del lenguaje C. 2.5 Identificadores y palabras reservadas. 2.6 Tipos de datos. 2.7 Constantes, variables y formación. 2.8 Declaraciones y expresiones. 2.8.1 Expresiones aritméticas. 2.8.2 Expresiones lógicas. 2.8.3 Expresiones relacionales. 2.9 Operadores.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			2.9.1 Operadores aritméticos. 2.9.2 Operadores relacionales y lógicos. 2.9.3 Operadores de asignación. 2.10 Instrucciones. 2.11 Entrada y salida de datos. 2.12 Funciones matemáticas. 3. Estructuras de control. 3.1 Estructura secuencial. 3.2 Estructura selectiva. 3.2.1 Estructura selectiva Simple. 3.2.2 Estructura selectiva Compuesta. 3.2.3 Estructura selectiva Múltiple. 3.2.4 Estructura selectivas anidadas. 3.3 Estructuras repetitivas. 3.3.1 Estructura repetitiva Mientras (while). 3.3.2 Estructura repetitiva Repite Mientras (do-while). 3.3.3 Estructura repetitiva Para (for). 3.3.4 Estructuras repetitivas anidadas. 4 Arreglos y cadenas. 4.1 Arreglos. 4.1.1 Declaración de arreglos. 4.1.2 Inicialización de arreglos. 4.1.3 Unidimensionales. 4.1.4 Bidimensionales. 4.1.5 Ordenamiento de arreglos. 4.1.6 Búsqueda en arreglos. 4.2 Cadenas de caracteres. 4.2.1 Biblioteca de manejo de caracteres.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			4.2.2 Funciones de conversión de cadenas. 5 Funciones. 5.1 Estructura de una función. 5.2 Acceso de una función. 5.3 Prototipo de funciones. 5.4 Paso de argumentos a una función. 6 Estructuras. 6.1 Declaración de una estructura. 6.2 Acceso a una estructura. 6.3 Estructuras anidadas. 6.4 Arreglos de estructuras. 6.5 Estructuras como parámetros.

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Automatizar el procesamiento de los datos generados en los procesos que así lo requiera utilizando algoritmos.		
Atributos (competencia específica) de la asignatura		
Diseñar e implementar algoritmos empleando el paradigma de Programación Estructurada para la solución de problemas utilizando el lenguaje C.		
Aportación a la competencia específica		Aportación a las competencias transversales
Saber	Saber hacer	Saber Ser
- Identificar las diferentes instrucciones de lenguaje C.	- Resolver problemas utilizando las diferentes estructuras del lenguaje C. - Abstraer, analizar y sintetizar la información de un contexto, para la resolución de problemas.	- Trabajar en forma autónoma y colaborativa.
Producto integrador de la asignatura, considerando los avances por unidad		
- Creación de un programa, dando solución a una necesidad real o simulada, incorporando las competencias desarrolladas en esta unidad de aprendizaje.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Etapas del diseño de un algoritmo para la automatización. de un proceso."

Número y nombre de la unidad: 1. Etapas del diseño de un algoritmo para la automatización. de un proceso.							
Tiempo y porcentaje para esta unidad:		Teoría:	7 horas	Práctica:	8 horas	Porcentaje del programa:	16.67%
Aprendizajes esperados: Identificar las etapas del diseño de un algoritmo para la automatización de un proceso.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1. Diseño de un programa y algoritmia. 1.1 Análisis de un problema. 1.2 Diseño. 1.3 Implementación y pruebas. 1.4 Algoritmia. 1.4.1 Definición de un problema. 1.4.2 Análisis de modelo matemático. 1.4.3 Desarrollo del algoritmo. 1.4.4 Diagrama de flujo. 1.4.5 Pseudocódigo. 1.4.6 Diagrama de flujo de datos. 1.4.7 Código.	Saber: - Definir qué es un algoritmo. - Describir los pasos para desarrollar un algoritmo. Saber hacer: - Aplicar los pasos para desarrollar un algoritmo. Ser: - Orden y disciplina.	- Explicación del tema por parte del docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo.	Evaluación diagnóstica: - Rescatar conocimientos previos con preguntas intercaladas. Evaluación formativa: - Elaboración de actividades como cuestionarios, resúmenes o mapas conceptuales. Evaluación sumativa: - Entrega de portafolio de evidencias.	Portafolio de evidencias: - Elaboración de un mapa conceptual o cuadro sinóptico o resumen de los conceptos abordados en la unidad de aprendizaje 1.			
Bibliografía							
- Corona, M. A.; Ancona, M. D. (2011). Diseño de algoritmos y su codificación en lenguaje C. Ciudad de México: McGraw Hill. - Deitel, P.; Deitel, H. (2014). Cómo Programar C. México: Pearson Educación. - Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana.							

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Introducción al lenguaje C."

Número y nombre de la unidad: 2. Introducción al lenguaje C.							
Tiempo y porcentaje para esta unidad:		Teoría:	7 horas	Práctica:	8 horas	Porcentaje del programa:	16.67%
Aprendizajes esperados:		Diseñar algoritmos utilizando operadores e instrucciones de salida, entrada y funciones matemáticas para la solución de un problema determinado.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
2. Introducción al lenguaje C. 2.1 Intérpretes y compiladores. 2.2 Estructura de un programa. 2.3 Programación estructurada. 2.4 Historia del lenguaje C. 2.5 Identificadores y palabras reservadas. 2.6 Tipos de datos. 2.7 Constantes, variables y formación. 2.8 Declaraciones y expresiones. 2.8.1 Expresiones aritméticas. 2.8.2 Expresiones lógicas. 2.8.3 Expresiones relacionales. 2.9 Operadores. 2.9.1 Operadores aritméticos. 2.9.2 Operadores relacionales y lógicos. 2.9.3 Operadores de asignación. 2.10 Instrucciones. 2.11 Entrada y salida de datos.	Saber: - Conocer la sintaxis del lenguaje C para el uso de operadores, entradas, salidas y funciones matemáticas. Saber hacer: - Diseñar algoritmos y su correspondiente codificación en lenguaje C utilizando operadores, entradas, salidas y funciones matemáticas. Ser: - Orden y disciplina. - Trabajo colaborativo.	- Explicación del tema por parte del docente con apoyo de recursos visuales. - Prácticas o resumen o mapa conceptual o investigación por equipo.	Evaluación formativa: - Elaboración de prácticas y actividades como cuestionarios, resúmenes o mapas conceptuales. Evaluación sumativa: - Entrega de portafolio de evidencias.	Portafolio de evidencias: - Prácticas realizadas en la unidad 2.			



Continuación: Tabla 4.2. Desglose específico de la unidad "Introducción al lenguaje C."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
2.12 Funciones matemáticas.				
Bibliografía				
<ul style="list-style-type: none">- Corona, M. A.; Ancona, M. D. (2011). Diseño de algoritmos y su codificación en lenguaje C. Ciudad de México: McGraw Hill.- Deitel, P.; Deitel, H. (2014). Cómo Programar C. México: Pearson Educación.- Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Estructuras de control."

Número y nombre de la unidad: 3. Estructuras de control.							
Tiempo y porcentaje para esta unidad:		Teoría:	7 horas	Práctica:	8 horas	Porcentaje del programa:	16.67%
Aprendizajes esperados: Diseñar algoritmos utilizando estructuras de control secuenciales, selectivas y repetitivas para la solución de un problema determinado.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
3. Estructuras de control. 3.1 Estructura secuencial. 3.2 Estructura selectiva. 3.2.1 Estructura selectiva Simple. 3.2.2 Estructura selectiva Compuesta. 3.2.3 Estructura selectiva Múltiple. 3.2.4 Estructura selectivas anidadas. 3.3 Estructuras repetitivas. 3.3.1 Estructura repetitiva Mientras (while). 3.3.2 Estructura repetitiva Repite Mientras (do-while). 3.3.3 Estructura repetitiva Para (for). 3.3.4 Estructuras repetitivas anidadas.	Saber: - Conocer la sintaxis del lenguaje C y la utilización de las estructuras de control secuenciales, selectivas y repetitivas. Saber hacer: -Diseñar algoritmos y su correspondiente codificación en lenguaje C utilizando las estructuras de control secuenciales, selectivas y repetitivas. - Abstracter la información de una situación planteada. Ser: - Orden y disciplina.	- Explicación del tema por parte del docente con apoyo de recursos visuales. - Prácticas o resumen o mapa conceptual o investigación por equipo.	Evaluación formativa: - Elaboración de prácticas y actividades como cuestionarios, resúmenes o mapas conceptuales. Evaluación sumativa: - Entrega de portafolio de evidencias.	Portafolio de evidencias: - Prácticas realizadas en la unidad 3.			



Continuación: Tabla 4.3. Desglose específico de la unidad "Estructuras de control."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	- Trabajo colaborativo.			

Bibliografía

- Corona, M. A.; Ancona, M. D. (2011). Diseño de algoritmos y su codificación en lenguaje C. Ciudad de México: McGraw Hill.
- Deitel, P.; Deitel, H. (2014). Cómo Programar C. México: Pearson Educación.
- Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.4. Desglose específico de la unidad "Arreglos y cadenas."

Número y nombre de la unidad: 4. Arreglos y cadenas.							
Tiempo y porcentaje para esta unidad:		Teoría:	7 horas	Práctica:	8 horas	Porcentaje del programa:	16.67%
Aprendizajes esperados:		Diseñar algoritmos utilizando arreglos y cadenas de caracteres para la solución de un problema determinado.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
4 Arreglos y cadenas. 4.1 Arreglos. 4.1.1 Declaración de arreglos. 4.1.2 Inicialización de arreglos. 4.1.3 Unidimensionales. 4.1.4 Bidimensionales. 4.1.5 Ordenamiento de arreglos. 4.1.6 Búsqueda en arreglos. 4.2 Cadenas de caracteres. 4.2.1 Biblioteca de manejo de caracteres. Funciones de conversión de cadenas.	Saber: - Conocer la sintaxis del lenguaje C para la manipulación de arreglos y cadenas de caracteres. Saber hacer: - Diseñar algoritmos y su correspondiente	- Explicación del tema por parte del docente con apoyo de recursos visuales. - Prácticas o resumen o mapa conceptual o investigación por equipo.	Evaluación formativa: - Elaboración de prácticas y actividades como cuestionarios, resúmenes o mapas conceptuales. Evaluación sumativa: - Entrega de portafolio de evidencias.	Portafolio de evidencias: - Prácticas realizadas en la unidad 4.			



Continuación: Tabla 4.4. Desglose específico de la unidad "Arreglos y cadenas."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	<p>codificación en lenguaje C utilizando arreglos y cadenas de caracteres.</p> <ul style="list-style-type: none">- Abstractar la información de una situación planteada. <p>Ser:</p> <ul style="list-style-type: none">- Orden y disciplina.- Trabajo colaborativo.			
Bibliografía				
<ul style="list-style-type: none">- Corona, M. A.; Ancona, M. D. (2011). Diseño de algoritmos y su codificación en lenguaje C. Ciudad de México: McGraw Hill.- Deitel, P.; Deitel, H. (2014). Cómo Programar C. México: Pearson Educación.- Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.5. Desglose específico de la unidad "Funciones."

Número y nombre de la unidad: 5. Funciones.							
Tiempo y porcentaje para esta unidad:		Teoría:	7 horas	Práctica:	8 horas	Porcentaje del programa:	16.67%
Aprendizajes esperados: Diseñar algoritmos utilizando el paradigma de programación modular para la solución de un problema determinado.							
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
5 Funciones. 5.1 Estructura de una función. 5.2 Acceso de una función. 5.3 Prototipo de funciones. 5.4 Paso de argumentos a una función.	Saber: - Conocer la sintaxis del lenguaje C para la utilización del paradigma de programación modular. Saber hacer: - Diseñar algoritmos y su correspondiente codificación en lenguaje C utilizando funciones definidas por el programador. - Abstractar la información de una situación planteada. Ser: - Orden y disciplina. - Trabajo colaborativo. - Analítico.	- Explicación del tema por parte del docente con apoyo de recursos visuales. - Prácticas o resumen o mapa conceptual o investigación por equipo. - Aprendizaje basado en proyecto.	Evaluación formativa: - Elaboración de prácticas y actividades como cuestionarios, resúmenes o mapas conceptuales. - Avance de proyecto. Evaluación sumativa: - Entrega de proyecto.	Elaboración de un proyecto final que contenga los temas de unidad 5 y 6.			
Bibliografía							
- Corona, M. A.; Ancona, M. D. (2011). Diseño de algoritmos y su codificación en lenguaje C. Ciudad de México: McGraw Hill. - Deitel, P.; Deitel, H. (2014). Cómo Programar C. México: Pearson Educación. - Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana.							

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.6. Desglose específico de la unidad "Estructuras."

Número y nombre de la unidad: 6. Estructuras.				
Tiempo y porcentaje para esta unidad:		Teoría: 7 horas	Práctica: 8 horas	Porcentaje del programa: 16.67%
Aprendizajes esperados: Diseñar algoritmos organizando la información en estructuras para la solución de problemas determinados.				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)
6 Estructuras. 6.1 Declaración de una estructura. 6.2 Acceso a una estructura. 6.3 Estructuras anidadas. 6.4 Arreglos de estructuras. 6.5 Estructuras como parámetros.	<p>Saber:</p> <ul style="list-style-type: none"> - Organizar la información en estructuras. <p>Saber hacer:</p> <ul style="list-style-type: none"> - Diseñar algoritmos y su correspondiente codificación en lenguaje C utilizando estructuras. - Abstractar la información de una situación planteada. <p>Ser:</p> <ul style="list-style-type: none"> - Orden y disciplina. - Trabajo colaborativo. - Analítico. 	<ul style="list-style-type: none"> - Explicación del tema por parte del docente con apoyo de recursos visuales. - Prácticas o resumen o mapa conceptual o investigación por equipo. - Aprendizaje basado en proyecto. 	<p>Evaluación formativa:</p> <ul style="list-style-type: none"> - Elaboración de prácticas y actividades como cuestionarios, resúmenes o mapas conceptuales. - Avance de proyecto. <p>Evaluación sumativa:</p> <ul style="list-style-type: none"> - Entrega de proyecto. 	Elaboración de un proyecto final que contenga los temas de unidad 5 y 6
Bibliografía				
<ul style="list-style-type: none"> - Corona, M. A.; Ancona, M. D. (2011). Diseño de algoritmos y su codificación en lenguaje C. Ciudad de México: McGraw Hill. - Deitel, P.; Deitel, H. (2014). Cómo Programar C. México: Pearson Educación. - Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana. 				



V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
Carrera(s): - Licenciatura en Informática.
- Licenciatura en Informática administrativa.
- Licenciatura en Informática y computación.
- Licenciatura en Ingeniería en Sistemas Computacionales.
- Licenciatura en Ingeniería en Desarrollo y Tecnologías del Software.
- Licenciatura en Ingeniería de Software.
- Ingeniería en Computación.
- Ingeniería en Informática.
- Ingeniería en Mecatrónica.
- Ingeniería en Sistemas de computación.
- Ingeniería en Sistemas computacionales.
- Ingeniería en Computación inteligente.
- Ingeniería en Desarrollo de Software.

- Ingeniería en Tecnología de Software.
- Ingeniería en Software.
- Ingeniería en Ciencias computacionales. o carrera afín
 - Experiencia profesional de mínimo un año de docente en materias a fines.
 - Experiencia mínima de dos años
 - Licenciatura.