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 orientada a objetos	
Academia:	Computación / Ciencias Sociales		**Clave:**	19SME04	
Módulo formativo:	Programación		**Seriación:**	- -	
Tipo de curso:	Presencial		**Prerrequisito:**	19SME03 - Programación estructurada	
Semestre:	Segundo	**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.	- Conocerá las diferentes metodologías asociadas a la identificación y solución de problemas aplicando punteros. - Conocerá los elementos que conforman el lenguaje de programación C++ mediante investigaciones o esquemas gráficos de sus antecedentes y componentes. - Identificará una metodología asertiva que permita aplicar los elementos de la Programación Orientada a Objetos para la resolución de problemas. - Identificará los aspectos metodológicos que permiten la comprensión de los fundamentos del diseño y estructura de una clase. - Conocerá los diferentes algoritmos de búsqueda y ordenamiento aplicables en la solución de problemas. - Identificará los diferentes algoritmos que conforman las estructuras de datos y el uso de archivos.	1. PUNTEROS. 1.1.Concepto de puntero. 1.2.Declaración de punteros. 1.3.Aritmética de direcciones. 1.4.Punteros y arreglos. 1.5.Punteros como parámetros. 1.6.Punteros como retorno de funciones. 2. EL LENGUAJE DE PROGRAMACIÓN C++ 2.1. Historia del lenguaje C++ 2.2. Funciones, palabras reservadas y comentarios. 2.3.Operadores C++, prioridad y orden de evaluación. 2.4.Conversión y tipo de datos. 2.5.Identificadores y declaraciones. 3. ENTRADA Y SALIDA ESTÁNDAR 3.1. Salida estándar y salida con formato. 3.2. Salida estándar de error. 3.3. Entrada estándar. 4. FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS. 4.1. Mecanismos básicos de la POO. 4.1.1. Objetos. 4.1.2. Mensajes. 4.1.3. Métodos. 4.1.4. Clase.

Continuación: Tabla 2. Objetivos educacionales (continuación			
No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			4.2. Características de la POO. 4.2.1. Abstracción. 4.2.2. Encapsulamiento. 4.2.3. Herencia. 4.2.4. Polimorfismo. 4.3. Comparación de la POO con la PE. 5. IMPLEMENTACIÓN DE CLASES. 5.1. Estructuras y clases. 5.2. Definición de una clase. 5.3. Miembro de una clase. 5.4. Datos miembro de una clase. 5.5. Funciones miembro de una clase. 5.6. Control de acceso a los miembros de una clase. 6. MÉTODOS DE ORDENAMIENTO Y BÚSQUEDA. 6.1. Algoritmos de ordenamiento directos. 6.2.Método de ordenamiento por intercambio. 6.3.Método de ordenamiento por selección. 6.4.Método de ordenamiento por inserción. 6.5.Método de ordenamiento por burbuja. 6.6.Algoritmos de ordenamientos indirectos. 6.7.Método de ordenamiento Shell. 6.8.Método de ordenamiento rápida (Quicksort). 6.9.Métodos de ordenamiento Binsort y Radixsort. 6.10.Métodos de búsqueda. 6.11.Búsqueda secuencial. 6.12.Búsqueda binaria. 7.ESTRUCTURAS DE DATOS.

Continuación: Tabla 2. Objetivos educacionales (continuación			
No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			7.1. Listas ligadas. 7.1.1. Concepto. 7.1.2. Creación. 7.1.3. Operaciones. 7.2.Listas circulares. 7.3. Listas doblemente ligadas. 7.4. Pilas. 7.5. Colas. 7.6. Árboles. 7.6.1. Concepto. 8. ARCHIVOS. 8.1. Entrada y salida de C++ 8.2. Manipulación de ficheros en el disco. 8.3. Acceso secuencial. 8.4. Acceso aleatorio. 8.5.Cadena de caracteres.
AE2	Desarrollar, innovar y/o implementar sistemas, procesos y productos para la solución integral de necesidades concretas en el campo de la mecatrónica.	- Aplicará los aspectos metodológicos que permiten la comprensión de los fundamentos del diseño y estructura de una clase. - Aplicará los diferentes algoritmos que conforman las estructuras de datos y el uso de archivos.	5. IMPLEMENTACIÓN DE CLASES. 5.7. Implementación de una clase. 5.8. Ámbito de una clase. 5.9. El puntero implícito this. 5.10. Funciones miembro y objetos constantes. 5.11. Inicialización de un objeto. 5.12. Constructor. 5.13. Asignación de objeto. 5.14. Destrucción de objetos. 5.15. Devolver static de una clase. 5.16. Punteros a miembros de una clase.

Continuación: Tabla 2. Objetivos educacionales (continuación			
No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
			5.17. Arreglos de objetos y puntero de objeto. 5.18. Funciones amigas de una clase. 5.19. Clases derivadas. 5.20. Clases base. 5.21. Herencia simple. 5.22. Constructores y destructores de clase. 5.23. Jerarquía de clases. 5.24. Punteros a objetos de una clase derivada. 5.25. Polimorfismo. 5.26. Herencia múltiple. 5.27. Operadores sobrecargados. 7. ESTRUCTURAS DE DATOS. 7.1. Listas ligadas. 7.1.1. Concepto. 7.1.2. Creación. 7.1.3. Operaciones. 7.2. Listas circulares. 7.3. Listas doblemente ligadas. 7.4. Pilas. 7.5. Colas. 7.6. Árboles. 7.6.1. Concepto. 8. ARCHIVOS. 8.1. Entrada y salida de C++ 8.2. Manipulación de ficheros en el disco. 8.3. Acceso secuencial. 8.4. Acceso aleatorio. 8.5. Cadena de caracteres.

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Optimizar los procesos mediante el modelado y desarrollo de programas de cómputo utilizando el paradigma orientado a objetos para la mejor comprensión de la información.		
Atributos (competencia específica) de la asignatura		
Diseñar e implementar soluciones mediante programas de cómputo, utilizando el paradigma orientado a objetos, con apoyo del lenguaje C++.		
Aportación a la competencia específica		**Aportación a las competencias transversales**
Saber	Saber hacer	Saber Ser
- Describir las características del paradigma orientado a objetos y conocer la sintaxis de su implementación en el lenguaje C++.	- Desarrollar programas de cómputo utilizando el paradigma orientado a objetos y alguna estructura dinámica, utilizando el lenguaje C++.	- Trabajo colaborativo. - Analítico, orden y disciplina.
Producto integrador de la asignatura, considerando los avances por unidad		
Proyecto integrador que involucre el desarrollo de un programa de cómputo que dé solución a una problemática de un contexto real, diseñando un modelo bajo el paradigma orientado a objetos, utilizando una o más estructuras de datos dinámicas y que el almacenamiento de información sea por medio de archivos, que permita demostrar la aplicación de las competencias desarrolladas en la unidad de aprendizaje.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Punteros."

Número y nombre de la unidad:	1. Punteros.					
Tiempo y porcentaje para esta unidad:	Teoría:	5 horas	Práctica:	10 horas	Porcentaje del programa:	16.67%
Aprendizajes esperados:	Comprender el uso de punteros (apuntadores) para el manejo de memoria en programas implementados en el lenguaje C.					

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
1. PUNTEROS. 1.1. Concepto de puntero. 1.2. Declaración de punteros. 1.3. Aritmética de direcciones. 1.4. Punteros y arreglos. 1.5. Punteros como parámetros. 1.6. Punteros como retorno de funciones.	Saber: - Definir qué es un puntero y la aritmética de direcciones de memoria. Saber hacer: - Declarar e implementar un puntero en el lenguaje C para el manejo de funciones, estructuras y arreglos unidimensionales. Ser: - Analítico, orden y disciplina.	- Explicación docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo. - Prácticas.	Evaluación formativa: - Prácticas. - Elaboración de cuestionarios escritos o cuestionario oral. - Actividades: resumen, mapa conceptual, investigación. Evaluación sumativa: - Entrega de portafolio.	Portafolio de evidencias: - Prácticas realizadas en la unidad. - Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 1.

Bibliografía

- 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.
- Sznajdleder, P. (2012). Algoritmos a fondo con implementaciones en C y Java. Argentina: Alfaomega.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "El lenguaje de programación C++"

Número y nombre de la unidad:	2. El lenguaje de programación C++				
Tiempo y porcentaje para esta unidad:	Teoría:	3 horas	Práctica:	5 horas	Porcentaje del programa: 8.89%
Aprendizajes esperados:	Identificar los principales elementos y su sintaxis del lenguaje C++, con el fin que desarrolle aplicaciones sencillas del lenguaje, aplicando la lógica matemática aprendida.				

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
2. EL LENGUAJE DE PROGRAMACIÓN C++ 2.1. Historia del lenguaje C++ 2.2. Funciones, palabras reservadas y comentarios 2.3. Operadores C++, prioridad y orden de evaluación 2.4. Conversión y tipo de datos 2.5. Identificadores y declaraciones	Saber: - Identificar los tipos de operadores, conversiones de tipos de datos, tipos de datos, variables y constantes en C++. Saber hacer: - Usar funciones, comentarios, operadores, tipos de datos, variables y constantes en el lenguaje C++. Ser: - Analítico, orden y disciplina.	- Explicación docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo. - Prácticas.	Evaluación formativa: - Prácticas. - Elaboración de cuestionarios escritos o cuestionario oral. - Actividades: resumen, mapa conceptual, investigación. Evaluación sumativa: - Entrega de portafolio.	Portafolio de evidencias: - Prácticas realizadas en la unidad. - Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 2.

Bibliografía

- 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.
- Sznajdleder, P. (2012). Algoritmos a fondo con implementaciones en C y Java. Argentina: Alfaomega.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Entrada y salida estándar."

Número y nombre de la unidad:	3. Entrada y salida estándar.					
Tiempo y porcentaje para esta unidad:	Teoría:	3 horas	Práctica:	4 horas	Porcentaje del programa:	7.78%
Aprendizajes esperados:	Identificar y comprender la sintaxis en el lenguaje C++ para las instrucciones de entrada y salida.					

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
3. ENTRADA Y SALIDA ESTÁNDAR. 3.1. Salida estándar y salida con formato. 3.2. Salida estándar de error. 3.3. Entrada estándar.	Saber: - Identificar la sintaxis de las instrucciones para entrada y salida de datos en el lenguaje C++. Saber hacer: - Declarar las instrucciones de entrada y salida de datos en el lenguaje C++ en la solución de problemas. Ser: - Analítico, orden y disciplina.	- Explicación docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo. - Prácticas.	Evaluación formativa: - Prácticas. - Elaboración de cuestionarios escritos o cuestionario oral. - Actividades: resumen, mapa conceptual, investigación. Evaluación sumativa: - Entrega de portafolio.	Portafolio de evidencias: - Prácticas realizadas en la unidad. - Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 3.

Bibliografía

- 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.
- Sznajdleder, P. (2012). Algoritmos a fondo con implementaciones en C y Java. Argentina: Alfaomega.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.4. Desglose específico de la unidad "Fundamentos de la programación orientada a objetos."

Número y nombre de la unidad:	4. Fundamentos de la programación orientada a objetos.					
Tiempo y porcentaje para esta unidad:	Teoría:	3 horas	Práctica:	7 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:	Comprender los fundamentos del paradigma orientado a objeto para la solución de problemas mediante el desarrollo de programas de cómputo.					

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
4. FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS. 4.1. Mecanismos básicos de la POO. 4.1.1. Objetos. 4.1.2. Mensajes. 4.1.3. Métodos. 4.1.4. Clase. 4.2. Características de la POO. 4.2.1. Abstracción. 4.2.2. Encapsulamiento. 4.2.3. Herencia. 4.2.4. Polimorfismo. 4.3. Comparación de la POO con la PE.	Saber: - Identificar y definir los principales conceptos del paradigma orientado a objetos. Saber hacer: - Implementar los conceptos del paradigma orientado a objetos en el Lenguaje de Modelado Universal (UML). Ser: - Analítico, orden y disciplina.	- Explicación docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo. - Ejercicios de modelado utilizando el paradigma orientado a objetos.	Evaluación formativa: - Prácticas. - Elaboración de cuestionarios escritos o cuestionario oral. - Actividades: resumen, mapa conceptual, investigación. Evaluación sumativa: - Entrega de portafolio.	Portafolio de evidencias: - Prácticas realizadas en la unidad. - Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 4.

Bibliografía

- 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.
- Sznajdleder, P. (2012). Algoritmos a fondo con implementaciones en C y Java. Argentina: Alfaomega.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.5. Desglose específico de la unidad "Implementación de clases."

Número y nombre de la unidad:	5. Implementación de clases.					
Tiempo y porcentaje para esta unidad:	Teoría:	3 horas	Práctica:	7 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:	Implementar en el lenguaje C++ el paradigma orientado a objetos para la solución de problemas mediante el desarrollo de programas de cómputo.					

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
5. IMPLEMENTACIÓN DE CLASES. 5.1. Estructuras y clases. 5.2. Definición de una clase. 5.3. Miembro de una clase. 5.4. Datos miembro de una clase. 5.5. Funciones miembro de una clase. 5.6. Control de acceso a los miembros de una clase. 5.7. Implementación de una clase. 5.8. Ámbito de una clase. 5.9. El puntero implícito this. 5.10. Funciones miembro y objetos constante. 5.11. Inicialización de un objeto. 5.12. Constructor. 5.13. Asignación de objeto. 5.14. Destrucción de objetos. 5.15. Devolver static de una clase. 5.16. Punteros a miembros de una clase.	Saber: - Conocer la sintaxis de implementación de clases y los conceptos del paradigma orientado a objetos en el lenguaje C++. Saber hacer: - Implementar en el lenguaje C++ el paradigma orientado a objetos para la solución de problemas. Ser: Analítico, orden y disciplina	- Explicación docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo. - Prácticas.	Evaluación formativa: - Prácticas. - Elaboración de cuestionarios escritos o cuestionario oral. - Actividades: resumen, mapa conceptual, investigación. Evaluación sumativa: - Entrega de portafolio.	Portafolio de evidencias: - Prácticas realizadas en la unidad. - Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 5.

Continuación: Tabla 4.5. Desglose específico de la unidad "Implementación de clases."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
5.17. Arreglos de objetos y puntero de objeto. 5.18. Funciones amigas de una clase. 5.19. Clases derivadas. 5.20. Clases base. 5.21. Herencia simple. 5.22. Constructores y destructores de clase. 5.23. Jerarquía de clases. 5.24. Punteros a objetos de una clase derivada. 5.25. Polimorfismo. 5.26. Herencia múltiple. 5.27. Operadores sobrecargados.				

Bibliografía

- 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.
- Sznajdleder, P. (2012). Algoritmos a fondo con implementaciones en C y Java. Argentina: Alfaomega.

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.6. Desglose específico de la unidad "Métodos de ordenamiento y búsqueda."

Número y nombre de la unidad:	6. Métodos de ordenamiento y búsqueda.					
Tiempo y porcentaje para esta unidad:	Teoría:	3 horas	Práctica:	7 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:	Comprender y aplicar los métodos de ordenamiento y búsqueda en el lenguaje C++ para la solución de problemas mediante el desarrollo de programas de cómputo.					

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
6. MÉTODOS DE ORDENAMIENTO Y BÚSQUEDA. 6.1. Algoritmos de ordenamiento directos. 6.2. Método de ordenamiento por intercambio. 6.3. Método de ordenamiento por selección. 6.4. Método de ordenamiento por inserción 6.5. Método de ordenamiento por burbuja. 6.6. Algoritmos de ordenamientos indirectos. 6.7. Método de ordenamiento Shell. 6.8. Método de ordenamiento rápida (Quicksort). 6.9. Métodos de ordenamiento Binsort y Radixsort. 6.10. Métodos de búsqueda. 6.11. Búsqueda secuencial.	Saber: - Comprender y describir el funcionamiento de los distintos métodos de ordenamiento y búsqueda Saber hacer: - Declarar y aplicar los métodos de ordenamiento y búsqueda en el lenguaje C++ utilizando arreglos unidimensionales. Ser: - Analítico, Orden y disciplina	- Explicación docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo. - Prácticas.	Evaluación formativa: - Prácticas. - Elaboración de cuestionarios escritos o cuestionario oral. - Actividades: resumen, mapa conceptual, investigación. Evaluación sumativa: - Entrega de portafolio.	Portafolio de evidencias: - Prácticas realizadas en la unidad. - Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 6.

Continuación: Tabla 4.6. Desglose específico de la unidad "Métodos de ordenamiento y búsqueda."				
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
6.12. Búsqueda binaria.				

Bibliografía

- Deitel, P., & Deitel, H. (2014). Cómo Programar C++. México: Person Educación.
- Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana.
- Sznajdleder,P. (2012). Algoritmos a fondo con implementaciones en C y Java. Argentina:Alfaomega

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.7. Desglose específico de la unidad "Estructura de datos."

Número y nombre de la unidad:	7. Estructura de datos.					
Tiempo y porcentaje para esta unidad:	Teoría:	7 horas	Práctica:	13 horas	Porcentaje del programa:	22.22%
Aprendizajes esperados:	Comprender e implementar las estructuras de datos bajo el paradigma orientado a objetos en el lenguaje C++ para la solución de problemas mediante el desarrollo de programas de cómputo.					

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
7. ESTRUCTURAS DE DATOS. 7.1. Listas ligadas. 7.1.1. Concepto. 7.1.2. Creación. 7.1.3. Operaciones. 7.2. Listas circulares. 7.3. Listas doblemente ligadas. 7.4. Pilas. 7.5. Colas. 7.6. Árboles. 7.6.1. Concepto.	Saber: - Definir y describir el funcionamiento de las estructuras de datos dinámicas (listas enlazadas, pilas, colas y árboles). Saber hacer: - Declarar y aplicar las estructuras de datos dinámicas en el lenguaje C++, utilizando el paradigma orientado a objetos. Ser: - Analítico, Orden y disciplina	- Explicación docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo. - Prácticas.	Evaluación formativa: - Prácticas. - Elaboración de cuestionarios escritos o cuestionario oral. - Actividades: resumen, mapa conceptual, investigación. Evaluación sumativa: - Entrega de portafolio.	Portafolio de evidencias: - Prácticas realizadas en la unidad. - Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 7.

Bibliografía

- Deitel, P., & Deitel, H. (2014). Cómo Programar C++. México: Person Educación.
- Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana.
- Sznajdleder,P. (2012). Algoritmos a fondo con implementaciones en C y Java. Argentina:Alfaomega

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.8. Desglose específico de la unidad "Archivos."

Número y nombre de la unidad:	8. Archivos.					
Tiempo y porcentaje para esta unidad:	Teoría:	3 horas	Práctica:	7 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:	Comprender e implementar el manejo de archivos en el lenguaje C++ para la solución de problemas mediante el desarrollo de programas de cómputo.					

Temas y subtemas (secuencia)	**Criterios de desempeño**	**Estrategias didácticas**	**Estrategias de evaluación**	**Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)**
8. ARCHIVOS. 8.1. Entrada y salida de C++ 8.2. Manipulación de ficheros en el disco. 8.3. Acceso secuencial. 8.4. Acceso aleatorio. 8.5. Cadena de caracteres.	Saber: - Definir y comprender el funcionamiento de archivos mediante el lenguaje C++. Saber hacer: - Declarar e implementar archivos en el lenguaje C++ para la solución de problemas. Ser: - Analítico, Orden y disciplina	- Explicación docente con apoyo de recursos visuales. - Resumen o mapa conceptual o investigación por equipo. - Prácticas.	Evaluación formativa: - Prácticas. - Elaboración de cuestionarios escritos o cuestionario oral. - Actividades: resumen, mapa conceptual, investigación. Evaluación sumativa: - Entrega de portafolio.	Portafolio de evidencias: - Prácticas realizadas en la unidad. - Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 8.

Bibliografía

- Deitel, P., & Deitel, H. (2014). Cómo Programar C++. México: Person Educación.
- Joyanes, L. (2008). Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos. España: McGraw Hill Interamericana.
- Sznajdleder,P. (2012). Algoritmos a fondo con implementaciones en C y Java. Argentina:Alfaomega

V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
Carrera(s): - Ingeniería en Computación. - Ingeniería en Sistemas. - Licenciatura en Computación. - Licenciatura en Tecnologías de la Información. - Ingeniería en Sistemas de computación. - Ingeniería en Computación inteligente. - Ingeniería en Sistemas computacionales. - Ingeniería en Ciencias computacionales. - Ingeniería de Software. - Ingeniería en Desarrollo de software. - Ingeniería en Tecnología de software. - Ingeniería en informática. - Ingeniería informática.

- Licenciatura en Ingeniería en desarrollo y tecnologías del software.

- Licenciatura en Informática y computación.

- Licenciatura en informática administrativa.

- Licenciatura en y diseño gráfico.

- Licenciatura en contaduría e informática.

o carrera afín

- Experiencia profesional de mínimo 1 año de manejo de algún lenguaje de programación orientado a objetos (C++, Java, PHP, entre otros).
- Experiencia mínima de dos años
- Licenciatura