



Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Marzo 31, 2022				
Carrera:	Ingeniería en Diseño Electrónico y Sistemas Inteligentes	Asignatura:	Programación orientada a objetos		
Academia:	Electrónica /	Clave:	19SDE04		
Módulo formativo:	Electrónica Digital	Seriación:	- -		
Tipo de curso:	Presencial	Prerrequisito:	19SDE01 - Programación estructurada		
Semestre:	Segundo	Créditos:	6.75	Horas semestre:	108 horas
Teoría:	2 horas	Práctica:	4 horas	Trabajo indpt.:	0 horas
				Total x semana:	6 horas

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
OE2	Los egresados implementarán proyectos especializados en sistemas complejos de control y electrónicos en organizaciones públicas o privadas.	Conocerán e implementarán las teorías de gestión y dirección aplicadas a proyectos.	50% de los egresados conocerán diferentes teorías de gestión y dirección de proyectos
OE3	Los egresados resolverán problemas en el ámbito industrial con el desarrollo de proyectos de sistemas electrónicos.	Conocerán e implementarán las metodologías de análisis y diseño de sistemas electrónicos.	30% de los egresados analizarán un sistema electrónico.
OE4	Los egresados se integrarán de manera satisfactoria en el ámbito laboral en las áreas de electrónica del sector público o privado.	Se integrarán al ámbito laboral a través de las estadías profesionales, trabajando de manera colaborativa en el desarrollo de proyectos.	30% de los egresados trabajarán de forma colaborativa en el desarrollo de proyectos en el sector público.
OE5	Los egresados aplicarán y administrarán sistemas electrónicos y de control de manera ética, con responsabilidad social para contribuir al desarrollo sustentable.	Conocerán e implementarán modelos de sistemas electrónicos y de control.	30% de los egresados aplicarán modelos de sistemas electrónicos o de control.
OE6	Los egresados se integrarán a redes de colaboración públicas o privadas para el desarrollo de proyectos tecnológicos nacionales e internacionales.	Se integrarán al trabajo colaborativo en instancias públicas (Conacyt) o privadas mediante las estadías, las materias de proyecto y el intercambio con otras instituciones.	30% de los egresados trabajarán de forma colaborativa en instancias públicas como Conacyt desarrollando proyectos.
OE1	Los egresados diseñarán y desarrollarán proyectos especializados en sistemas complejos de control y electrónicos en organizaciones públicas o privadas.	Conocerán y aplicarán la metodología de la formulación, diseño, implementación y evaluación de Proyectos de tipo Industrial y de tecnologías Electrónicas Emergentes.	40% de los Egresados serán capaces de formular proyectos Electrónicos.



Atributos de egreso de plan de estudios		Criterios de desempeño	Componentes
AE1	Aplicar los conocimientos de ciencias básicas, como la química, física y matemáticas, y las ciencias de la ingeniería para resolver problemas dentro del campo de la electrónica.	- Reconocerá el alcance de la programación orientada a objetos, mediante el desarrollo completo de programas de temáticas diversas.	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. 2.1 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. 3.1 Salida estándar y salida con formato. 3.2 Salida estándar de error. 3.3 Entrada estándar. 4.1 Mecanismos básicos de la POO. 4.2 Objetos. 4.3 Mensajes. 4.4 Métodos. 4.5 Características de la POO. 4.6 Abstracción. 4.7 Encapsulamiento. 4.8 Herencia. 4.9 Polimorfismo. 4.10 Comparación de la POO con la PE.
			5.1 Estructuras y clases. 5.2 Definición de una clase.



Continuación: Tabla 2. Objetivos educativos (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
AE3	Implementar estrategias a partir del juicio ingenieril para sacar conclusiones y tomar decisiones a partir de análisis estadísticos y mejorar así la calidad de los procesos industriales.	- Conocerá las bases para la resolución de problemas, mediante la formulación y el desarrollo de pasos específicos, orientados a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora.	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 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. 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.
		- Conocerá las nuevas aplicaciones en la programación orientada a objetos, así como el uso necesario en el desarrollo de nueva tecnología.	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.



Continuación: Tabla 2. Objetivos educacionales (continuación)

No.	Atributos de egreso de plan de estudios	Criterios de desempeño	Componentes
AE6	Reconocer la mejora continua como parte de su desarrollo profesional para diseñar e implementar sistemas analógicos y/o digitales y resolver problemas dentro del campo de la electrónica.		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. 7.1 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. 8.1 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, utilizando el 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 de programas de cómputo utilizando el paradigma orientado a objetos y alguna estructura dinámica, utilizando el lenguaje C++.	Abstraer, analizar y sintetizar la información de un contexto, para la resolución de problemas. 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 se solución a una problemática de un contexto real: Diseño de un modelo bajo el paradigma orientado a objetos, utilizando una o más estructura de datos dinámica y el almacenamiento de información sea por medio de archivos, que permita demostrar la aplicación de las competencias desarrolladas por 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:	13.89%
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)			
1.1. 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: Abstractar, analizar y sintetizar la información de un contexto, para la resolución de problemas. Analítico, Orden y disciplina.	- Resumen. - Mapa conceptual. - Investigación por equipo. - Prácticas.	Evaluación formativa: - Actividades y ejercicios. - Prácticas. Evaluación Sumativa: - Cuestionarios escritos o cuestionario oral.	Portafolio de evidencias de las 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). <i>Cómo Programar C++</i> . México: Person Educación. - Joyanes, L. (2008). <i>Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos</i> . España: McGraw Hill Interamericana. - Sznajdleder ,P. (2012). <i>Algoritmos a fondo con implementaciones en C y Java</i> . 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:	15 horas	Porcentaje del programa:	16.67%
Aprendizajes esperados: Identificar los principales elementos y su sintaxis del 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)			
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: - Declarar funciones, comentarios, operadores, tipos de datos, variables y constantes en el lenguaje C++. Ser: Abstraer, analizar y sintetizar la información de un contexto, para la resolución de problemas. Analítico, Orden y disciplina.	- Resumen. - Mapa conceptual. - Investigación por equipo. - Prácticas.	Evaluación formativa: - Actividades y ejercicios. - Prácticas. Evaluación Sumativa: - Cuestionarios escritos o cuestionario oral.	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). <i>Cómo Programar C++</i> . México: Person Educación. - Joyanes, L. (2008). <i>Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos</i> . España: McGraw Hill Interamericana. - Sznajdleder ,P. (2012). <i>Algoritmos a fondo con implementaciones en C y Java</i> . 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:	7 horas	Porcentaje del programa:	9.26%
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)			
3.1 Salida estándar y salida con formato. 3.2 Salida estándar de error. 3.3 Entrada estándar.	<p>Saber:</p> <ul style="list-style-type: none"> - Identificar la sintaxis de las instrucciones para entrada y salida de datos en el lenguaje C++. <p>Saber hacer:</p> <ul style="list-style-type: none"> - Declarar las instrucciones de entrada y salida de datos en el lenguaje C++ en la solución de problemas. <p>Ser:</p> <ul style="list-style-type: none"> - Abstractar, analizar y sintetizar la información de un contexto, para la resolución de problemas. - Analítico, Orden y disciplina. 	<ul style="list-style-type: none"> - Resumen. - Mapa conceptual. - Investigación por equipo. - Prácticas. 	<p>Evaluación formativa:</p> <ul style="list-style-type: none"> - Actividades y ejercicios. - Prácticas. <p>Evaluación Sumativa:</p> <ul style="list-style-type: none"> - Cuestionarios escritos o cuestionario oral. 	Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 3.			
Bibliografía							
<ul style="list-style-type: none"> - Deitel, P., & Deitel, H. (2014). <i>Cómo Programar C++</i>. México: Person Educación. - Joyanes, L. (2008). <i>Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos</i>. España: McGraw Hill Interamericana. - Sznajdleder, P. (2012). <i>Algoritmos a fondo con implementaciones en C y Java</i>. 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:	9.26%
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)			
4.1 Mecanismos básicos de la POO. 4.2 Objetos. 4.3 Mensajes. 4.4 Métodos. 4.5 Características de la POO. 4.6 Abstracción. 4.7 Encapsulamiento. 4.8 Herencia. 4.9 Polimorfismo. 4.10 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: Abstractar, analizar y sintetizar la información de un contexto, para la resolución de problemas. Analítico, Orden y disciplina.	- Resumen. - Mapa conceptual. - Investigación por equipo. - Ejercicios de modelado utilizando el paradigma orientado a objetos.	Evaluación formativa: - Actividades y ejercicios. - Prácticas. Evaluación Sumativa: - Cuestionarios escritos o cuestionario oral.	Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 4. Portafolio de evidencias de los ejercicios realizados en la unidad.			
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 4.5. 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:	4 horas	Práctica:	9 horas	Porcentaje del programa:	12.04%
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)			
5.1 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 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. 5.17 Arreglos de objetos y puntero de objeto.	Saber: - Conocer la sintaxis de implementación de clases y los conceptos del paradigma orientado a objetos en el lenguaje C++.	- Resumen. - Mapa conceptual. - Investigación por equipo. - Prácticas.	Evaluación formativa: - Actividades y ejercicios. - Prácticas. Evaluación Sumativa: - Cuestionarios escritos o cuestionario oral.	Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 5. Portafolio de evidencias de las prácticas realizadas en la unidad.			



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
5.18 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				
<ul style="list-style-type: none">- Deitel, P., & Deitel, H. (2014). <i>Cómo Programar C++</i>. México: Person Educación.- Joyanes, L. (2008). <i>Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos</i>. España: McGraw Hill Interamericana.- Sznajdleder ,P. (2012). <i>Algoritmos a fondo con implementaciones en C y Java</i>. 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: 9.26%
Aprendizajes esperados:		Comprender e implementar 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)
6.1 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.	Saber: - Comprender y describir el funcionamiento de los distintos métodos de ordenamiento y búsqueda Saber hacer: - Declarar e implementar los métodos de ordenamiento y búsqueda en el lenguaje C++ utilizando arreglos unidimensionales. Ser: Abstraer, analizar y sintetizar la información de un contexto, para la resolución de problemas. Analítico, Orden y disciplina.	- Resumen. - Mapa conceptual. - Investigación por equipo. - Prácticas.	Evaluación formativa: - Actividades y ejercicios. - Prácticas. Evaluación Sumativa: - Cuestionarios escritos o cuestionario oral.	Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 6. Portafolio de evidencias de las prácticas realizadas en la unidad.



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

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:	18.52%
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)			
7.1 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 e implementar las estructuras de datos dinámicos en el lenguaje C++, utilizando el paradigma orientado a objetos. Ser: Abstractar, analizar y sintetizar la información de un contexto, para la resolución de problemas. Analítico, Orden y disciplina.	- Resumen. - Mapa conceptual. - Investigación por equipo. - Prácticas.	Evaluación formativa: - Actividades y ejercicios. - Prácticas. Evaluación Sumativa: - Cuestionarios escritos o cuestionario oral.	Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 7. Portafolio de evidencias de las prácticas realizadas en la unidad.			
Bibliografía							
- Deitel, P., & Deitel, H. (2014). <i>Cómo Programar C++</i> . México: Person Educación. - Joyanes, L. (2008). <i>Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos</i> . España: McGraw Hill Interamericana. - Sznajdleder, P. (2012). <i>Algoritmos a fondo con implementaciones en C y Java</i> . 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: 9 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)
8.1 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.	<p>Saber:</p> <ul style="list-style-type: none"> - Definir y comprender el funcionamiento de archivos mediante el lenguaje C++. <p>Saber hacer:</p> <ul style="list-style-type: none"> - Declarar e implementar archivos en el lenguaje C++ para la solución de problemas. <p>Ser:</p> <ul style="list-style-type: none"> - Abstractar, analizar y sintetizar la información de un contexto, para la resolución de problemas. - Analítico, Orden y disciplina. 	<ul style="list-style-type: none"> - Resumen. - Mapa conceptual. - Investigación por equipo. - Prácticas. 	<p>Evaluación formativa:</p> <ul style="list-style-type: none"> - Actividades y ejercicios. - Prácticas. <p>Evaluación Sumativa:</p> <ul style="list-style-type: none"> - Cuestionarios escritos o cuestionario oral. 	<p>Elaboración de un resumen o mapa conceptual o investigación de los conceptos abordados en la unidad de aprendizaje 8.</p> <p>Portafolio de evidencias de las prácticas realizadas en la unidad.</p>
Bibliografía				
<ul style="list-style-type: none"> - Deitel, P., & Deitel, H. (2014). <i>Cómo Programar C++</i>. México: Person Educación. - Joyanes, L. (2008). <i>Fundamentos de Programación: Algoritmos y Estructuras de datos y objetos</i>. España: McGraw Hill Interamericana. - Sznajdleder, P. (2012). <i>Algoritmos a fondo con implementaciones en C y Java</i>. Argentina: Alfaomega. 				



V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
<p>Carrera(s): - Ingeniería en Computación.</p> <ul style="list-style-type: none">- Ingeniería en Sistemas.- Licenciatura en Computación.- Licenciatura en Tecnologías de la Información o Carrera afín. <p>o carrera afín</p> <ul style="list-style-type: none">- 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 en el área de conocimiento. Preferentemente Maestría relacionada con el área de conocimiento.