



Programa de asignatura por competencias de educación superior

Sección I. Identificación del Curso

Tabla 1. Identificación de la Planificación del Curso.

Actualización:	Junio 07, 2022				
Carrera:	Ingeniería en Desarrollo de Software	Asignatura:	Programación estructurada		
Academia:	Ciencias Computacionales y Programación /	Clave:	19SDS02		
Módulo formativo:	Programación aplicada	Seriación:	19SDS06 - Programación orientada a objetos		
Tipo de curso:	Presencial	Prerrequisito:	- -		
Semestre:	Primero	Créditos:	10.13	Horas semestre:	162 horas
Teoría:	2 horas	Práctica:	4 horas	Trabajo indpt.:	3 horas
				Total x semana:	9 horas

Sección II. Objetivos educacionales

Tabla 2. Objetivos educacionales

Objetivos educacionales		Criterios de desempeño	Indicadores
OE1	Los egresados gestionarán recursos relacionados con el desarrollo de software en alguna organización.	Los egresados podrán aplicar metodologías en el desarrollo de proyectos en el contexto laboral.	20% de los egresados aplicarán metodologías en el desarrollo de software en su contexto laboral.
OE2	Los egresados diseñarán e implementarán soluciones innovadoras mediante el uso de tecnologías de la información.	Los egresados participarán activamente en el ciclo de desarrollo e integración continuos	25% de los egresados desempeñarán labores de desarrollo e integración continuos.
OE5	Los egresados serán capaces de emprender un negocio basado en el desarrollo de un producto o servicio de tecnologías de la información, aportando valor a la generación de empleos e incrementar el bienestar económico y social, de forma ecológica y sustentable.	Los egresados serán capaces de emprender un negocio basado en el desarrollo propio de un producto o servicio de tecnologías de la información.	2% de los egresados tendrán participación en el acta constitutiva de una empresa creada a partir del desarrollo de software para ofrecer un producto o servicio.
Atributos de egreso de plan de estudios		Criterios de desempeño	Componentes
AE1	Aplicar los conocimientos de ciencias básicas como física y matemáticas, así como las ciencias de la ingeniería para generar nuevos productos o servicios basándose en la innovación tecnológica.	Diseñará aplicaciones en lenguaje C que cumplan con los principios de diseño de la ingeniería de software.	1. Introducción a la programación. 2. Lenguaje C, Introducción. 3. Estructuras de control en lenguaje C.
AE3	Desarrollar una experimentación adecuada para recopilar, almacenar y analizar grandes cantidades de información basándose en el juicio ingenieril para crear productos o servicios innovadores mediados por software.	Fragmentará un problema general, de tal modo que cada fragmento se pueda resolver mediante una subrutina o función.	4.-Estructuras básicas de datos 5.-Punteros 6.-Funciones 7.-archivos de texto o binarios 8.-operaciones bit a bit

Sección III. Atributos de la asignatura

Tabla 3. Atributos de la asignatura

Problema a resolver		
Analizar, generar y codificar algoritmos para automatizar algún proceso de evaluación de datos.		
Atributos (competencia específica) de la asignatura		
Ser capaz de proponer, entender y analizar un algoritmo que resuelva alguna necesidad en cuanto a procesamiento de datos se refiera.		
Aportación a la competencia específica		Aportación a las competencias transversales
Saber	Saber hacer	Saber Ser
<ul style="list-style-type: none"> - Conocer los tipos de datos que se pueden llegar a usar en una aplicación de software en lenguaje C. - Conocer las estructuras de control que se pueden llegar a aplicar en el desarrollo de software. 	<ul style="list-style-type: none"> - Analizar problemas de la vida cotidiana para encontrar un algoritmo que lo represente. - Implementar algoritmos en lenguaje C para resolver problemas de índole cotidiana. 	Debido a que en la actualidad las aplicaciones son desarrolladas por más de una persona se requiere que se trabaje de manera colaborativa, lo cual implica que el alumno debe tener una comunicación efectiva con sus compañeros de trabajo, ser responsable con su trabajo y los compromisos adquiridos como miembro de un equipo.
Producto integrador de la asignatura, considerando los avances por unidad		
El alumno debe desarrollar un proyecto que se encargue de gestionar los datos vertidos en un archivo, de tal modo que dicho archivo solo debe contener los datos personales de sus compañeros de grupo.		

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.1. Desglose específico de la unidad "Conceptos generales de la programación estructurada."

Número y nombre de la unidad: 1. Conceptos generales de la programación estructurada.							
Tiempo y porcentaje para esta unidad:		Teoría:	10 horas	Práctica:	6 horas	Porcentaje del programa:	14.81%
Aprendizajes esperados:		Entender y aplicar los conceptos de la programación estructurada para poder decidir qué tipo de paradigma de programación sería más conveniente usar en ciertas situaciones.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
1 Introducción a la Programación. 1.1 Conceptos generales. 1.1.1 Qué es la programación. 1.1.2 Algoritmos. 1.1.3 Seudocódigo. 1.1.4 Técnicas de programación en este contexto. 1.2 Lenguajes de programación. 1.2.1 Lenguaje máquina. 1.2.2 Lenguaje ensamblador. 1.2.3 Lenguaje de alto nivel. 1.2.4 Proceso de compilación de un lenguaje compilado. 1.3 Características de un programa estructurado. 1.3.1 Características generales. 1.3.2 Tipos de estructuras de control. 1.3.3 Tipos de datos. 1.3.4 Variables y constantes.	Saber: - Discernir entre los distintos tipos de datos informáticos que pueden intervenir en una aplicación de software. Saber hacer: - Proponer una o varias soluciones a través de algoritmos y pseudocódigos. - Estructurar un programa de acuerdo al algoritmo desarrollado.	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación. - Debate sobre los distintos tipos de datos que se usan en la vida cotidiana, lo cual lleve a generar una retroalimentación colaborativa. - Analogías de la vida cotidiana en la resolución de problemas y los procesos para resolver problemas computacionalmente. - Casos de uso para ejemplificar la	Evaluación formativa: - Ejercicios. - Evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. Evaluación sumativa: - Exámenes, entre escritos y orales.	Portafolio de evidencias con los ejercicios realizados en la unidad.			



Continuación: Tabla 4.1. Desglose específico de la unidad "Conceptos generales de la programación estructurada."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
1.3.5 Estructuración por funciones. 1.4 Codificación numérica. 1.4.1 Concepto de código. 1.4.2 Los números como sistema de codificación de cantidades. 1.4.3 Fundamentos matemáticos para un sistema de numeración. 1.4.4 Bases, dígitos y cifras. 1.4.5 Sistema Binario. 1.4.6 Cambio de Base. 1.4.7 Complementos a	Ser: - El alumno podrá transmitir sus ideas de manera asertiva a otros miembros del equipo de trabajo. - Trabajo colaborativo. - Responsabilidad y compromiso.	importancia de la programación y los distintos elementos que intervienen en el desarrollo de un programa.		
Bibliografía				
- Alcover, P.M. (2012). Informática aplicada. Programación en lenguaje C. Universidad Politécnica de Cartagena. - Ortega, B. (2011). Programando en C a Bajo Nivel. Universidad de Buenos Aires Facultad de Ingeniería.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.2. Desglose específico de la unidad "Lenguaje C, Introducción."

Número y nombre de la unidad: 2. Lenguaje C, Introducción.							
Tiempo y porcentaje para esta unidad:		Teoría:	2 horas	Práctica:	12 horas	Porcentaje del programa:	12.96%
Aprendizajes esperados:		Conocer los distintos operadores y tipos de datos en la programación, para asegurar que los datos, dentro de la aplicación, no se corrompan por el manejo de diferentes TIPOS de datos de las diferentes variables que sean usadas en la aplicación.					
Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad (Evidencia de aprendizaje de la unidad)			
2 Lenguaje C, Introducción. 2.1 Introducción. 2.1.2 Entorno de programación. 2.1.3 Estructura básica de un programa en C. 2.1.4 Elementos léxicos. 2.1.5 Sentencias simples y sentencias compuestas. 2.1.6 Errores y depuración. 2.2 Tipos de datos, operadores y variables. 2.2.1 Tipos de variables y constantes. 2.2.2 Tipos de datos y su declaración 2.2.3 Operadores relacionales, aritméticos, binarios, especiales, lógicos y de asignación, su funcionalidad y uso. 2.2.4 Conversión forzada de tipos. 2.2.5 Alcance de las variables. 2.3 Funciones de biblioteca. 2.3.1 Entrada y salida por consola.	Saber: - Conocer los distintos tipos de operadores en la resolución de un problema específico. Saber hacer: - Establecer las variables y constantes que pudieran llegarse a necesitar en la codificación de un algoritmo previamente desarrollado. - Usar los distintos tipos de operadores en la resolución de un problema específico.	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación. - Plantear un problema o situación en la cual se pueda aplicar la programación y que dé como resultado la solución del problema en cuestión. - Para corroborar los aprendizajes se puede hacer uso de un aula invertida.	Evaluación formativa: - Ejercicios. - Evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. Evaluación sumativa: - Exámenes, entre escritos y orales.	Portafolio de evidencias con los ejercicios realizados en la unidad.			



Continuación: Tabla 4.2. Desglose específico de la unidad "Lenguaje C, Introducción."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
2.3.2 Funciones matemáticas. 2.3.3 Funciones para comparación.	Ser: <ul style="list-style-type: none">- El alumno podrá transmitir sus ideas de manera asertiva a otros miembros del equipo de trabajo.- Trabajo colaborativo.- Responsabilidad y compromiso.			
Bibliografía				
<ul style="list-style-type: none">- Alcover, P.M. (2012). Informática aplicada. Programación en lenguaje C. Universidad Politécnica de Cartagena.- Ortega, B. (2011). Programando en C a Bajo Nivel. Universidad de Buenos Aires Facultad de Ingeniería.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.3. Desglose específico de la unidad "Estructuras de control en lenguaje C."

Número y nombre de la unidad: 3. Estructuras de control en lenguaje C.				
Tiempo y porcentaje para esta unidad:		Teoría: 3 horas	Práctica: 12 horas	Porcentaje del programa: 13.89%
Aprendizajes esperados:		Conocer las distintas formas de estructuras de control en la programación, para generar aplicaciones capaces de analizar conjuntos grandes de datos de manera automática.		
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 en lenguaje C. 3.1 De tipo condicional de selección. 3.1.1 if, bifurcación abierta. 3.1.2 else if, bifurcación cerrada. 3.1.3 switch, selección múltiple. 3.2 De tipo repetitivas. 3.2.1 for, hacer hasta. 3.2.2 while, mientras hacer. 3.2.3 do-while, hacer mientras. 3.2.4 break y continue.	Saber: - Conocer las distintas formas de estructuras de control en la programación. - Nombrar las distintas estructuras de control que existen en lenguaje C. Saber hacer: - Nombrar las distintas estructuras de control que existen en lenguaje C. - Utilizar la mejor estructura de control, a su juicio, para resolver alguna parte del algoritmo que se esté codificando.	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación que se lleven a cabo como los de tarea. - Plantear un problema o situación en la cual se pueda aplicar la programación y quedé como resultado la solución del problema en cuestión.	Evaluación formativa: - Ejercicios. - Evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. Evaluación sumativa: - Exámenes, entre escritos y orales.	Portafolio de evidencias con los ejercicios realizados en la unidad.



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

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - El alumno podrá transmitir sus ideas de manera asertiva a otros miembros del equipo de trabajo. - Trabajo colaborativo. - Responsabilidad y compromiso.	- Para corroborar los aprendizajes se puede hacer uso de un aula invertida.		
Bibliografía				
- Alcover, P.M. (2012). Informática aplicada. Programación en lenguaje C. Universidad Politécnica de Cartagena. - Ortega, B. (2011). Programando en C a Bajo Nivel. Universidad de Buenos Aires Facultad de Ingeniería.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.4. Desglose específico de la unidad "Estructuras básicas de datos."

Número y nombre de la unidad: 4. Estructuras básicas de datos.							
Tiempo y porcentaje para esta unidad:		Teoría:	3 horas	Práctica:	10 horas	Porcentaje del programa:	12.04%
Aprendizajes esperados:		Conocer los distintos de estructuras de datos básicas en la programación, con el fin de poder almacenar, en la memoria temporal, bajo un mismo nombre a todo un conjunto de datos, él cual será analizado después de haber sido almacenado.					
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 Arreglos. 4.1.1 Arreglos numéricos unidimensionales y multidimensionales. 4.1.2 Arreglos de caracteres unidimensionales y multidimensionales. 4.1.3 Enumeraciones. 4.2 Estructuras y tipos definidos. 4.2.1 Definición de una estructura. 4.2.2 Arreglo de estructuras. 4.2.3 Definición y uso de Uniones. 4.2.4 Registros REGS.	Saber: - Conocer las distintas estructuras de datos básicas para ayudar en la resolución del algoritmo en cuestión. Saber hacer: - Analizar, de acuerdo a su experiencia, algoritmos que le sean adecuados para el análisis de los datos contenidos en el arreglo. - Usar las distintas estructuras de datos básicas para ayudar en la resolución del algoritmo en cuestión.	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación que se lleven a cabo como los de tarea. - Plantear un problema o situación en la cual se pueda aplicar la programación y quedé como resultado la solución del problema en cuestión.	Evaluación formativa: - Ejercicios. - Evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. Evaluación sumativa: - Exámenes, entre escritos y orales.	Portafolio de evidencias con los ejercicios realizados en la unidad.			



Continuación: Tabla 4.4. Desglose específico de la unidad "Estructuras básicas de datos."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - El alumno podrá transmitir sus ideas de manera asertiva a otros miembros del equipo de trabajo. - Trabajo colaborativo. - Responsabilidad y compromiso.	- Para corroborar los aprendizajes se puede hacer uso de un aula invertida.		
Bibliografía				
- Alcover, P.M. (2012). Informática aplicada. Programación en lenguaje C. Universidad Politécnica de Cartagena. - Ortega, B. (2011). Programando en C a Bajo Nivel. Universidad de Buenos Aires Facultad de Ingeniería.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.5. Desglose específico de la unidad "Punteros y asignación dinámica de memoria."

Número y nombre de la unidad: 5. Punteros y asignación dinámica de memoria.				
Tiempo y porcentaje para esta unidad:		Teoría: 2 horas	Práctica: 12 horas	Porcentaje del programa: 12.96%
Aprendizajes esperados:		Comprender y usar el mecanismo de los datos referenciados, lo cual hace que se haga un uso eficiente de los recursos de hardware con que disponga el sistema 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 Punteros de tipos primitivos. 5.1.1 Definición de lo que es un puntero. 5.1.2 Declaración de punteros. 5.1.3 Asignación de memoria dinámica para datos primitivos y definidos.	Saber: - Conocer los punteros como parte de una mejor administración de los recursos de hardware. - Comprender en qué ocasiones es mejor usar referencias que copias de datos. Saber hacer: - Usar los punteros como parte de una mejor administración de los recursos de hardware. - Usar punteros como mecanismo para eficientar el desempeño del programa en cuestión.	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación que se lleven a cabo como los de tarea. - Plantear un problema o situación en la cual se pueda aplicar la programación y que dé como resultado la solución del problema en cuestión. - Para corroborar los aprendizajes se puede hacer uso de un aula invertida.	Evaluación formativa: - Ejercicios. - Evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. Evaluación sumativa: - Exámenes, entre escritos y orales.	Portafolio de evidencias con los ejercicios realizados en la unidad.



Continuación: Tabla 4.5. Desglose específico de la unidad "Punteros y asignación dinámica de memoria."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	Ser: - El alumno podrá transmitir sus ideas de manera asertiva a otros miembros del equipo de trabajo. - Trabajo colaborativo. - Responsabilidad y compromiso.			
Bibliografía				
- Alcover, P.M. (2012). Informática aplicada. Programación en lenguaje C. Universidad Politécnica de Cartagena. - Ortega, B. (2011). Programando en C a Bajo Nivel. Universidad de Buenos Aires Facultad de Ingeniería.				

Sección IV. Desglose específico por cada unidad formativa

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

Número y nombre de la unidad: 6. Funciones.							
Tiempo y porcentaje para esta unidad:		Teoría:	2 horas	Práctica:	10 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:		Ser capaz de diseñar y usar funciones, tanto de biblioteca, como las elaboradas propias, para eficientar procesos que se repitan constantemente dentro del algoritmo diseñado, además de generar pequeñas tareas (funciones) que en conjunto resuelven el problema en general.					
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 Definición de funciones. 6.1.1 Diseño de funciones. 6.1.2 Declaraciones y prototipos. 6.1.3 Sentencia return. 6.1.4 Parámetros en una función. 6.2 Llamadas a funciones. 6.2.1 Llamadas por valor. 6.2.2 Llamadas por referencia. 6.2.3 Ámbito de variables.	Saber: - Comprender el mecanismo de estructurar su programa en base a las funciones que él mismo diseñe. Saber hacer: - Usar funciones de biblioteca que mejor ayuden en la resolución del problema. - Diseñar sus propias funciones de acuerdo con el algoritmo a resolver. Ser: - El alumno podrá transmitir sus ideas de manera asertiva a otros miembros del equipo de trabajo. - Trabajo colaborativo. - Responsabilidad y compromiso.	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación que se lleven a cabo como los de tarea. - Plantear un problema o situación en la cual se pueda aplicar la programación y quedé como resultado la solución del problema en cuestión.	Evaluación formativa: - Ejercicios. - Evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. Evaluación sumativa: - Exámenes, entre escritos y orales.	Portafolio de evidencias con los ejercicios realizados en la unidad.			



Continuación: Tabla 4.6. Desglose específico de la unidad "Funciones."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
		- Para corroborar los aprendizajes se puede hacer uso de un aula invertida.		
Bibliografía				
- Alcover, P.M. (2012). Informática aplicada. Programación en lenguaje C. Universidad Politécnica de Cartagena. - Ortega, B. (2011). Programando en C a Bajo Nivel. Universidad de Buenos Aires Facultad de Ingeniería.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.7. Desglose específico de la unidad "Gestión de archivos."

Número y nombre de la unidad: 7. Gestión de archivos.							
Tiempo y porcentaje para esta unidad:		Teoría:	2 horas	Práctica:	10 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:		Usar archivos binarios o de texto como parte de la solución a un problema/ algoritmo en el cual sea necesario tener almacenados datos de manera permanente y no de manera volátil, y que dichos datos puedan ser recuperados en un futuro mediante el acceso a dicho archivo.					
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 Archivos de texto. 7.1.1 Definición de archivo de texto. 7.1.2 Tratamiento de archivos en lenguaje C. 7.1.3 Funciones para gestión de archivos de texto. 7.2 Archivos binarios. 7.2.1 Definición de archivo de binarios. 7.2.2 Tratamiento de archivos binarios en lenguaje C. 7.2.3 Funciones para gestión de archivos de binarios.	Saber: - Entender el proceso de la gestión de archivos binarios y de texto. Saber hacer: - Usar los archivos como fuente de datos, en cualquier parte del proceso CRUD. - Elegir el tipo de archivo que mejor convenga usar, de acuerdo al tipo de datos que se generan en la aplicación. Ser: - El alumno podrá transmitir sus ideas de manera asertiva a otros miembros del	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación que se lleven a cabo como los de tarea. - Plantear un problema o situación en la cual se pueda aplicar la programación y que dé como resultado la solución del problema en cuestión. - Para corroborar los aprendizajes se puede hacer uso de un aula invertida.	Evaluación formativa: - Ejercicios. - Evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. Evaluación sumativa: - Exámenes, entre escritos y orales.	Portafolio de evidencias con los ejercicios realizados en la unidad.			



Continuación: Tabla 4.7. Desglose específico de la unidad "Gestión de archivos."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	equipo de trabajo. - Trabajo colaborativo. - Responsabilidad y compromiso.			
Bibliografía				
- Alcover, P.M. (2012). Informática aplicada. Programación en lenguaje C. Universidad Politécnica de Cartagena. - Ortega, B. (2011). Programando en C a Bajo Nivel. Universidad de Buenos Aires Facultad de Ingeniería.				

Sección IV. Desglose específico por cada unidad formativa

Tabla 4.8. Desglose específico de la unidad "Programación a bajo nivel en lenguaje C."

Número y nombre de la unidad: 8. Programación a bajo nivel en lenguaje C.							
Tiempo y porcentaje para esta unidad:		Teoría:	2 horas	Práctica:	10 horas	Porcentaje del programa:	11.11%
Aprendizajes esperados:		Aplicar los diferentes mecanismos de análisis de datos, pero a nivel de bits, con el fin de poder generar algoritmos de encriptación, aplicaciones hardware/software.					
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 Operaciones bit a bit. 8.1.1 Operación AND bit a bit. 8.1.2 Operación OR bit a bit. 8.1.3 Operación X OR bit a bit. 8.2 Corrimiento de bits. 8.2.1 Corrimientos a la izquierda. 8.2.2 Corrimiento a la derecha. 8.2.3 Complemento a 1 de un número.	Saber: - Entender cómo hacer operaciones booleanas de datos a nivel de bits. Saber hacer: - Usar diferentes operaciones y mecanismo de análisis a nivel de bits, de acuerdo al problema que se quiera resolver. - Determinar en qué situaciones es indispensable usar operaciones binarias tanto para el análisis como para la transformación de los datos. Ser: - El alumno podrá transmitir sus ideas de	- Cada participante debe generar su portafolio de evidencias, tanto de las notas personales como de los distintos ejercicios de programación que se lleven a cabo como los de tarea. - Plantear un problema o situación en la cual se pueda aplicar la programación y que dé como resultado la solución del problema en cuestión. - Para corroborar los aprendizajes se puede hacer uso de un aula invertida.	Evaluación formativa: - Ejercicios. - Evaluaciones orales podrán abordar preguntas específicas de sus trabajos entregados en el portafolio de evidencias. Evaluación sumativa: - Exámenes, entre escritos y orales.	Portafolio de evidencias con los ejercicios realizados en la unidad.			



Continuación: Tabla 4.8. Desglose específico de la unidad "Programación a bajo nivel en lenguaje C."

Temas y subtemas (secuencia)	Criterios de desempeño	Estrategias didácticas	Estrategias de evaluación	Producto Integrador de la unidad
	manera asertiva a otros miembros del equipo de trabajo. - Trabajo colaborativo. - Responsabilidad y compromiso.			
Bibliografía				
- Alcover, P.M. (2012). Informática aplicada. Programación en lenguaje C. Universidad Politécnica de Cartagena. - Ortega, B. (2011). Programando en C a Bajo Nivel. Universidad de Buenos Aires Facultad de Ingeniería.				



V. Perfil docente

Tabla 5. Descripción del perfil docente

Perfil deseable docente para impartir la asignatura
<p>Carrera(s): - Ingeniería en Tecnología de la Información y Comunicación.</p> <ul style="list-style-type: none">- Licenciatura en informática.- Ingeniería en Tecnologías de la Información y Comunicación.- Ingeniería en Tecnologías de la Información y Comunicaciones- Ingeniería en Tecnologías de 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. <p>o carrera afín</p> <ul style="list-style-type: none">- Con experiencia docente o en el campo deseable de 2 años. Manejo de TIC's. Con habilidades pedagógicas y uso de metodologías alternativas de enseñanza.

- Experiencia mínima de dos años
- Licenciatura o superior.