

1. Datos de identificación:

Nombre de la institución	Universidad Autónoma de Nuevo León
Nombre de la dependencia	Facultad de Ciencias Físico Matemáticas
Nombre del programa educativo	Licenciatura en Ciencias Computacionales/ Licenciatura en Multimedia y Licenciatura en ciencias computacionales
Nombre de la unidad de aprendizaje	Programación Estructurada
Horas aula-teoría y/o práctica, totales	75
Frecuencias aula por semana	5
Horas extra aula, totales	45
Tipo de modalidad	Escolarizada
Tipo de periodo académico	2 semestre
Tipo de unidad de aprendizaje	Obligatoria
Área curricular	ACFP-F
Créditos UANL	4
Fecha de elaboración	04/12/17
Fecha de última actualización	04/12/17
Responsable(s) del diseño y actualización:	MIA. Ernesto Solís Valenzuela, LCC. Jorge Humberto Solís Rodríguez.

2. Presentación:

Ésta unidad de aprendizaje se desarrolla en cuatro etapas: la primera etapa es fundamentos de lenguaje C, inicia con la comprensión de las características del lenguaje C, identifica las bibliotecas estándar del lenguaje C, los tipos de datos y expresiones para luego analizar la estructura básica de un programa en lenguaje C, y la entrada y salida de datos, concluyendo ésta etapa con el diseño de programas secuenciales.

La segunda etapa es Estructuras de selección y repetición, comienza con el diseño de programas con estructuras de selección, concluyendo con el diseño de programas con estructuras de repetición.

La tercera etapa es arreglos y estructuras, donde identificamos los diferentes tipos de arreglos, diseñamos programas utilizando arreglos de una y dos dimensiones, y realizamos programas aplicando estructuras para el almacenamiento de datos en memoria.

En la cuarta etapa funciones, punteros y archivos, aplicamos la programación modular mediante la manipulación de funciones, y el intercambio de datos entre ellas, así mismo utilizamos punteros para un manejo más eficiente de la memoria y almacenamos permanentemente la información en archivos de texto y binario.

El desarrollo de estas cuatro etapas propicia en el estudiante un aprendizaje significativo y por ende el logro de las competencias planteadas en el programa analítico, dando pie al desarrollo del producto integrador de aprendizaje, el cual consiste en un proyecto de desarrollo de una aplicación empresarial o científico, un documento ejecutivo digital y la presentación oral del mismo.

Esta unidad de aprendizaje introduce al estudiante a la Programación, permitiéndole obtener las habilidades necesarias para analizar, diseñar e implementar programas, desarrollando programas reales y comprobar el funcionamiento correcto.

3. Propósito(s):

Esta unidad de aprendizaje enseña al estudiante a identificar conceptos, estructuras lógicas y las técnicas de programación para el desarrollo de soluciones a problemas de procesamiento de información. Esta unidad de aprendizaje promueve en el estudiante habilidades para analizar, diseñar y codificar programas estructurados con el lenguaje de programación C para implementar soluciones a problemas computacionales

Esta unidad de aprendizaje requiere que el estudiante conozca los conceptos de algoritmo, diagrama de flujo y pseudocódigo, impartidos en la unidad de aprendizaje Metodología de la Programación y aplique razonamiento lógico para la solución de problemas. Esta unidad de aprendizaje proporciona al estudiante nuevos conceptos, estructuras de control, técnicas y la lógica de un lenguaje de programación estructurada, fácil de entender y aplicar, que aporta las bases para comprender la implementación de la programación orientada a objetos de la unidad de aprendizaje Programación Orientada a Objetos y de la unidad de aprendizaje estructura de datos.

Al finalizar esta unidad de aprendizaje permitirá al estudiante identificar cada estructura del lenguaje de programación para crear soluciones de tecnologías de información, reconociendo fácilmente el código por medio del cual se transmite los diversos mensajes de la información, los datos y los resultados. El estudiante dominará el código del lenguaje de programación. Así mismo, el estudiante se acepta como miembro de un equipo interdisciplinario, respetando las diferencias que surgen de la interacción humana al realizar trabajo en equipo para proyectos de desarrollo de sistemas computacionales. Los estudiantes del equipo generan ambientes y proyectos de inclusión para el éxito del proyecto.

Al desarrollar un proyecto de tecnologías de información el estudiante identifica necesidades y reconoce correctamente la interacción entre los elementos que generan estas necesidades, genera y valora diferentes alternativas de solución, identifica el

impacto en la organización, propone, desarrolla e implementa la solución seleccionada y valida la solución a través de pruebas piloto.

Esta unidad de aprendizaje le proporciona al estudiante una herramienta computacional para la creación de software, permitiendo comparar con las diversas herramientas del mercado, para el desarrollo de tecnologías de información en todos los niveles de las organizaciones y en el ámbito científico.

4. Competencias del perfil de egreso:

Competencias generales a las que contribuye esta unidad de aprendizaje

Instrumentales

2. Utilizar los lenguajes lógico, formal, matemático, icónico, verbal y no verbal de acuerdo a su etapa de vida, para comprender, interpretar y expresar ideas, sentimientos, teorías y corrientes de pensamiento con un enfoque ecuménico.

Personales y de Interacción Social

9. Mantener una actitud de compromiso y respeto hacia la diversidad de prácticas sociales y culturales que reafirman el principio de integración en el contexto local, nacional e internacional con la finalidad de promover ambientes de convivencia pacífica.

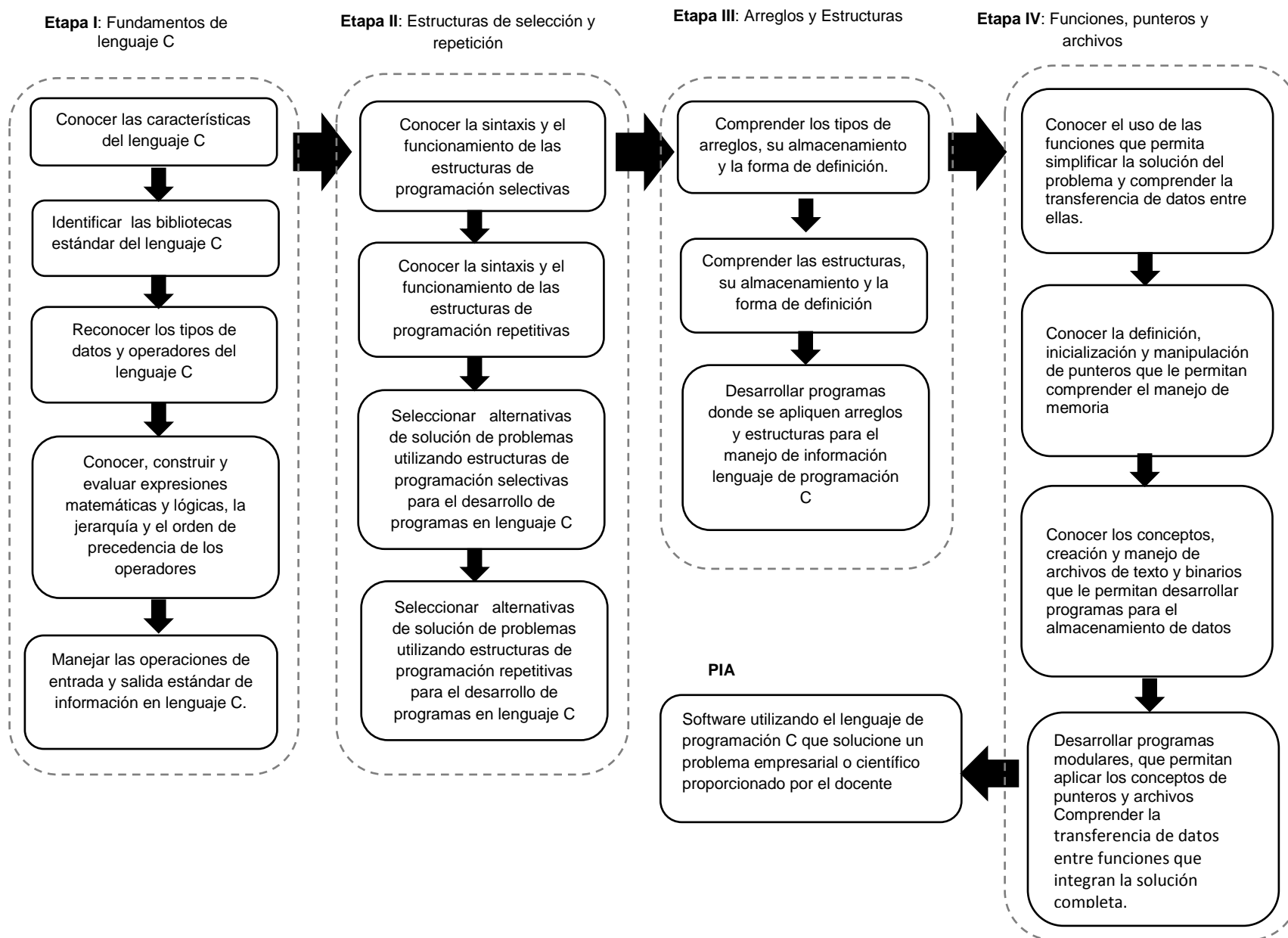
Integradoras

12. Construir propuestas innovadoras basadas en la comprensión holística de la realidad para contribuir a superar los retos del ambiente global interdependiente.

Competencias específicas del perfil de egreso a las que contribuye la unidad de aprendizaje

3. Crear software utilizando herramientas computacionales de última generación, con el fin de incrementar la competitividad de las organizaciones mediante la automatización de procesos.

5. Representación gráfica:



6. Estructuración en etapas o fases:

Etapa I: Fundamentos de lenguaje C

Etapa II: Estructuras de selección y repetición

Etapa III: Arreglos y Estructuras

Etapa IV: Funciones, punteros y archivos

Etapa I: Fundamentos de lenguaje C

Elemento de competencia:

- Comprender los fundamentos, bibliotecas estándar, tipos de datos y operadores del lenguaje C para construir la estructura de un programa en lenguaje c.
- Aplicar estructuras secuenciales mediante la programación en lenguaje C para la solución de un conjunto de ejercicios y problemas comunes.

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos
1. Evaluación teórico de fundamentos del lenguaje C.	<ul style="list-style-type: none"> Respetar el horario. Responde de forma individual Presenta las soluciones de forma escrita. Responde de forma clara y concisa. Resuelve la evaluación mediante la plataforma Nexus 	<ul style="list-style-type: none"> El profesor realiza el encuadre de la Unidad de Aprendizaje. El profesor inicia una actividad de integración con la finalidad de que todos los estudiantes se conozcan. El profesor realiza 	1.1. Historia del Lenguaje C 1.2. Características del lenguaje C. 1.3. ANSI C estándar 1.4. Palabras claves de C	<ul style="list-style-type: none"> Equipo de cómputo. Proyector Libro de texto: Joyanes Aguilar, L., & Zahonero Martínez, I. (2005). Programación en C, metodologías, estructura de datos y objetos. Plataforma Nexus.

		<p>una actividad de diagnóstico de conceptos computacionales.</p> <ul style="list-style-type: none"> • El profesor expone los fundamentos del lenguaje C, y la evolución de equipo de cómputo. • El estudiante investiga la historia de del desarrollo del lenguaje C y crea un mapa conceptual. (Actividad 1.1 ponderada) 		
2. Evaluación práctica, tipos de datos y operadores del lenguaje C y estructuras secuenciales	<ul style="list-style-type: none"> • Respeta el horario. • Resuelve de forma individual • Desarrolla de forma escrita la evaluación. • Presentar elaborada la actividad 1.3 • No se permite el uso de notas de apoyo ni formularios. • Incluye el código para cada problema. • Responde de forma clara y concisa. • Resuelve la evaluación durante la 	<ul style="list-style-type: none"> • El profesor expone las bibliotecas estándar, tipos de datos y operadores del lenguaje C • El profesor desarrolla ejemplos de problemas, donde se apliquen los diferentes tipos de datos y operadores, para construir y evaluar expresiones matemáticas y lógicas, la jerarquía y el orden de 	<p>2.1. Tipos de datos de C.</p> <p>2.2. Modificadores const y volatile.</p> <p>2.3. Conversiones de tipos de datos.</p> <p>2.4. Operadores.</p> <p>2.5. Bibliotecas estándar de C.</p> <p>2.6. Declaración de variables y constantes.</p> <p>2.7. Forma general de un programa en C.</p> <p>2.8. Errores más</p>	<ul style="list-style-type: none"> • Equipo de cómputo. • Proyector • Libro de texto: Joyanes Aguilar, L., & Zahonero Martínez, I. (2005). Programación en C, metodologías, estructura de datos y objetos. • Plataforma Nexus.

	sesión de clase.	<p>precedencia de los operadores</p> <ul style="list-style-type: none"> • El estudiante participará en la solución de ejemplos propuestos por el maestro, será elegido al azar. • El estudiante resuelve de manera individual, una serie de 10 ejercicios sobre expresiones algebraicas y lógicas. (Actividad 1.2 ponderada). • El profesor expone las operaciones de entrada y salida estándar de información en lenguaje C. • El profesor desarrolla ejemplos de problemas, donde se apliquen las operaciones de entrada y salida de información en el lenguaje C. • El estudiante participará en la 	comunes.	
--	------------------	---	----------	--

		<p>solución de ejemplos propuestos por el maestro, será elegido al azar.</p> <ul style="list-style-type: none"> El estudiante resuelve de manera individual, una serie de 10 ejercicios sobre las operaciones de entrada y salida de información en el lenguaje C. (Actividad 1.3 ponderada). 		
--	--	--	--	--

Etapla II: Estructuras de selección y repetición

Elemento de competencia: a) Aplicar estructuras condicionales, selectivas y repetitivas mediante la programación en lenguaje C para la solución de un conjunto de ejercicios y problemas comunes

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos
3. Evaluación práctica de solución de problemas que incluyan estructuras de selección y de repetición.	<ul style="list-style-type: none"> Respeto el horario. Resuelve de forma individual Desarrolla de forma escrita la evaluación. Presentar elaborada la actividad 2.1 No se permite el uso de notas de apoyo ni formularios. Responde de forma 	<ul style="list-style-type: none"> El profesor expone los tipos de estructuras condicionales, selectivas y repetitivas. El profesor desarrolla ejemplos de solución de problemas, donde se apliquen los diferentes tipos de 	<p>3.1. Estructuras de control.</p> <p>3.2. Expresiones lógicas.</p> <p>3.3. Estatuto de control if.</p> <p>3.4. Estatuto if anidado.</p> <p>3.5. Estatuto de control switch.</p> <p>3.6. Operador</p>	<ul style="list-style-type: none"> Equipo de cómputo. Proyector Libro de texto: Joyanes Aguilar, L., & Zahonero Martínez, I. (2005). Programación en C, metodologías, estructura de datos y objetos. Plataforma Nexus

	<p>clara y concisa.</p> <ul style="list-style-type: none"> • Resuelve la evaluación durante la sesión de clase. • La solución presentada por el alumno deberá incluir el análisis, diseño y código en lenguaje C. • La solución deberá ser realizada en un tiempo máximo de 2 horas. • Deberá resolver un mínimo de 2 problemas diferentes. 	<p>estructuras condicionales, selectivas y repetitivas.</p> <ul style="list-style-type: none"> • El estudiante participará en la solución de ejemplos propuestos por el maestro, será elegido al azar. • El estudiante resuelve ejercicios de condición, selección y repetición (Actividad 2.1 ponderada). 	<p>condicional.</p> <p>3.7. Errores más comunes</p> <p>3.8. Ciclos en programas: while, for y do/while.</p> <p>3.9. Ciclos de conteo y ciclos condicionales.</p> <p>3.10. Diseño de ciclos.</p> <p>3.11. Ciclos anidados.</p> <p>3.12. Sentencias break y continue.</p> <p>3.13. Depuración y prueba de programas.</p> <p>3.14. Errores más comunes.</p>	
--	---	--	--	--

Etapa III: Arreglos y Estructuras

Elementos de competencias: Aplicar arreglos y estructuras para la solución de un conjunto de ejercicios y problemas mediante la programación en lenguaje C.

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos
4. Evaluación práctica de arreglos y estructuras.	<ul style="list-style-type: none"> • Respetar el horario. • Presentar la evaluación de forma individual, escrita y/o en plataforma Nexus. • Presentar elaborada la actividad 3.1 • No se permite el uso de notas de apoyo ni formularios • Responder de forma clara y concisa. • Resolver la evaluación durante la sesión de clase la cual debe ser realizada en un tiempo máximo de 2 horas. • La solución presentada por el alumno deberá incluir el análisis, diseño y código en lenguaje C. 	<ul style="list-style-type: none"> • El profesor expone los temas de declaración e inicialización de arreglos de una dimensión y dos dimensiones su almacenamiento en memoria. • El profesor desarrolla ejemplos de solución de problemas, donde se apliquen arreglos de una dimensión. • El estudiante participará en la solución de ejemplos de arreglos de una dimensión propuestos por el maestro, será elegido al azar. • El profesor expone el concepto de arreglo de dos o más dimensiones. 	<p>4.1. Declaración e inicialización de un arreglo.</p> <p>4.2. Arreglos de los diferentes tipos de datos básicos.</p> <p>4.3. Tipo de dato struct (estructura).</p> <p>4.4. Arreglos de tipo struct.</p>	<ul style="list-style-type: none"> • Equipo de cómputo. • Proyector • Libro de texto: Joyanes Aguilar, L., & Zahonero Martínez, I. (2005). Programación en C, metodologías, estructura de datos y objetos. • Plataforma Nexus

	<ul style="list-style-type: none"> • Deberá resolver un mínimo de 2 problemas con dificultad baja y media. 	<ul style="list-style-type: none"> • El profesor resuelve problemas implementando arreglos de dos dimensiones • El estudiante participará en la solución de ejemplos de arreglos de 2 o más dimensiones propuestos por el maestro, será elegido al azar. • El profesor explica mediante la solución de problemas la implementación de estructuras de datos. • El estudiante participará en la solución de ejemplos de estructuras propuestos por el maestro, será elegido al azar. • El estudiante resuelve el laboratorio de arreglos y estructuras (Actividad 3.1 ponderada) 		
--	---	---	--	--

Etapas IV: Funciones, punteros y archivos

Elementos de competencias:

- Desarrollar programas en lenguaje C utilizando funciones para simplificar el diseño optimizando la solución de problemas y la transferencia de datos
- Manejar punteros mediante programas de lenguaje C que permitan entender el manejo de memoria.
- Aplicar el manejo de archivos en el desarrollo de aplicaciones para el almacenamiento de datos.

Evidencias de aprendizaje	Criterios de desempeño	Actividades de aprendizaje	Contenidos	Recursos
5. Evaluación práctica de arreglos y estructuras.	<ul style="list-style-type: none"> Respetar el horario. Presentar la evaluación de forma individual, escrita y/o en plataforma Nexus. Presentar elaborada la actividad 4.1 No se permite el uso de notas de apoyo ni formularios Responder de forma clara y concisa. Resolver la evaluación durante la sesión de clase la cual debe ser realizada en un tiempo máximo de 2 horas. La solución presentada por el alumno deberá incluir el análisis, diseño y código en 	<ul style="list-style-type: none"> El profesor expone los temas de diseño de soluciones Top Down. El profesor expone el tema de funciones, tipos, prototipos, paso de argumentos. El profesor desarrolla ejemplos de solución de problemas, donde se apliquen los contenidos de funciones. El estudiante participará en la solución de ejemplos de funciones propuestos por el maestro, será elegido al azar. El profesor expone el concepto de punteros, su 	<ol style="list-style-type: none"> Diseño de soluciones tipo Top. Down. Diseño paso a paso con funciones. Tipos de Funciones Prototipo de funciones Arreglos como argumentos de funciones. Errores más comunes Declaración de variables de tipo puntero. Inicialización de punteros. Punteros a arreglos. Aritmética de punteros. 	<ul style="list-style-type: none"> Equipo de cómputo. Proyector Libro de texto: Joyanes Aguilar, L., & Zahonero Martínez, I. (2005). Programación en C, metodologías, estructura de datos y objetos. Plataforma Nexus

	<p>lenguaje C.</p> <ul style="list-style-type: none"> • Deberá resolver un mínimo de 2 problemas con dificultad baja y media. 	<p>declaración, inicialización y su manejo de memoria. dimensiones.</p> <ul style="list-style-type: none"> • El profesor resuelve problemas implementando punteros. • El estudiante participará en la solución de ejemplos de propuestos por el maestro, será elegido al azar. • El profesor explica mediante la solución de problemas la utilización de archivos de texto y binarios. • El estudiante participará en la solución de archivos de texto y binarios propuestos por el profesor, será elegido al azar. • El estudiante resuelve ejercicios de funciones, punteros y archivo (Actividad 4.1 ponderada) 	<p>5.11. Punteros a funciones</p> <p>5.12. Introducción a archivos.</p> <p>5.13. Diferencias entre los tipos de archivos.</p> <p>5.14. Funciones definidas por C para escribir y leer un solo carácter en un archivo de texto.</p> <p>5.15. Funciones para escribir y leer una cadena de caracteres en un archivo de texto</p>	
--	--	---	--	--

7. Evaluación integral de procesos y productos:

Etapas I: Algoritmos y diagramas de Flujo

Actividad de aprendizaje 1.1	2%
Actividad de aprendizaje 1.2	2%
Actividad de aprendizaje 1.3	2%
Evidencia de aprendizaje 1	10%

Etapas II: Estructuras de condición y repetición

Actividad de aprendizaje 2.1	4%
Evidencia de aprendizaje 2	10%

Etapas III: Estructuras de datos

Actividad de aprendizaje 3.1	5%
Evidencia de aprendizaje 3	20%

Etapas IV: Funciones, punteros y archivos

Actividad de aprendizaje 4.1	5%
Evidencia de aprendizaje 4	20%

Producto Integrador de Aprendizaje 20%

Total 100%

8. Producto integrador de aprendizaje:

Software utilizando el lenguaje de programación C que solucione un problema empresarial o científico proporcionado por el docente, entregando un documento ejecutivo digital y presentado en forma oral.

9. Fuentes de apoyo y consulta:

Cairó Battistutti, O. (2006). Fundamentos de programación, piensa en C (Primera ed.). Person Educación.

Deitel, H. M., & Deitel, P. J. (2004). Como programar en C/C++ (Cuarta ed.). México: Prentice Hall.

Joyanes Aguilar, L., & Zahonero Martínez, I. (2005). Programación en C, metodologías, estructura de datos y objetos. México: McGraw Hill.

López Román, L. (2005). Programación Estructurada en Lenguaje C (Segunda ed.). Alfa Omega.

Programación en Castellano, S. (2017). Programación web. Obtenido de El portal de programación en castellano: <http://programacion.net/>

cplusplus.com. (2017). Referencia del lenguaje C/C++. Obtenido de <http://www.cplusplus.com/>

elsevier.com. (2017). Base de datos SCOPUS. Obtenido de <http://www.scopus.com>