



1. DATOS GENERALES DE LA UNIDAD DE APRENDIZAJE (UA) O ASIGNATURA			
Nombre de la Unidad de Aprendizaje (UA) o Asignatura			Clave de la UA
SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE PROGRAMACIÓN			15883
Modalidad de la UA	Tipo de UA	Área de formación	Valor en créditos
Escolarizada	Seminario	Básica común	5
UA de pre-requisito	UA simultáneo	UA posteriores	
Ninguna	Se sugiere cursar Programación	INF - Estructura de Datos I (15888) COM - Estructura de Datos I (15888) Para las carreras de BIM y CEL no tiene seriación.	
Horas totales de teoría	Horas totales de práctica	Horas totales del curso	
0	68	68	
Licenciatura(s) en que se imparte		Módulo al que pertenece	
Ingeniería en Computación Ingeniería en Informática Ingeniería en Comunicaciones y Electrónica Ingeniería Biomédica		Arquitectura y Programación de Sistemas Sistemas de Información Electrónica Digital Instrumentación Médica	
Departamento		Academia a la que pertenece	
Ciencias Computacionales		Programación	
Elaboró		Fecha de elaboración o revisión	
Ruben Núñez Ortega Sonia Osorio Angel		junio de 2023	



2. DESCRIPCIÓN DE LA UA O ASIGNATURA	
Presentación	
Este seminario aporta al perfil de los Ingenieros en Informática, Ingenieros en Computación, Ingenieros Biomédicos e Ingenieros en Comunicaciones y Electrónica los conocimientos, habilidades, metodología, así como capacidades de análisis y síntesis, para plantear la solución de problemas susceptibles de ser computarizados, utilizando el lenguaje C por ser un lenguaje de programación Estructurada.	
Relación con el perfil	
Modular	De egreso
Esta UA pertenece a los módulos de Arquitectura y Programación de Sistemas, Sistemas de Información, Electrónica Digital, e Instrumentación Médica, cuyo propósito es la aplicación de elementos de programación estructurada para dar solución a problemas fundamentales de un sistema de cómputo. Esta UA ayuda a la consecución de dicho propósito al codificar algoritmos de solución a problemas sencillos, utilizando como herramienta el lenguaje C.	La UA aporta al perfil de egreso de: Ingeniería en Computación: Las bases para el diseño y desarrollo de sistemas de software. Ingeniería Informática: Las habilidades para desarrollar algoritmos y su codificación. Ingeniería en Comunicaciones y Electrónica: Las bases planificar, diseñar, implementar y verificar proyectos que incluyan sistemas digitales mediante el uso de lenguajes de descripción de hardware de alto nivel. Ingeniería Biomédica: desarrollar y mejorar herramientas o técnicas de detección, diagnóstico, tratamiento, pronóstico.



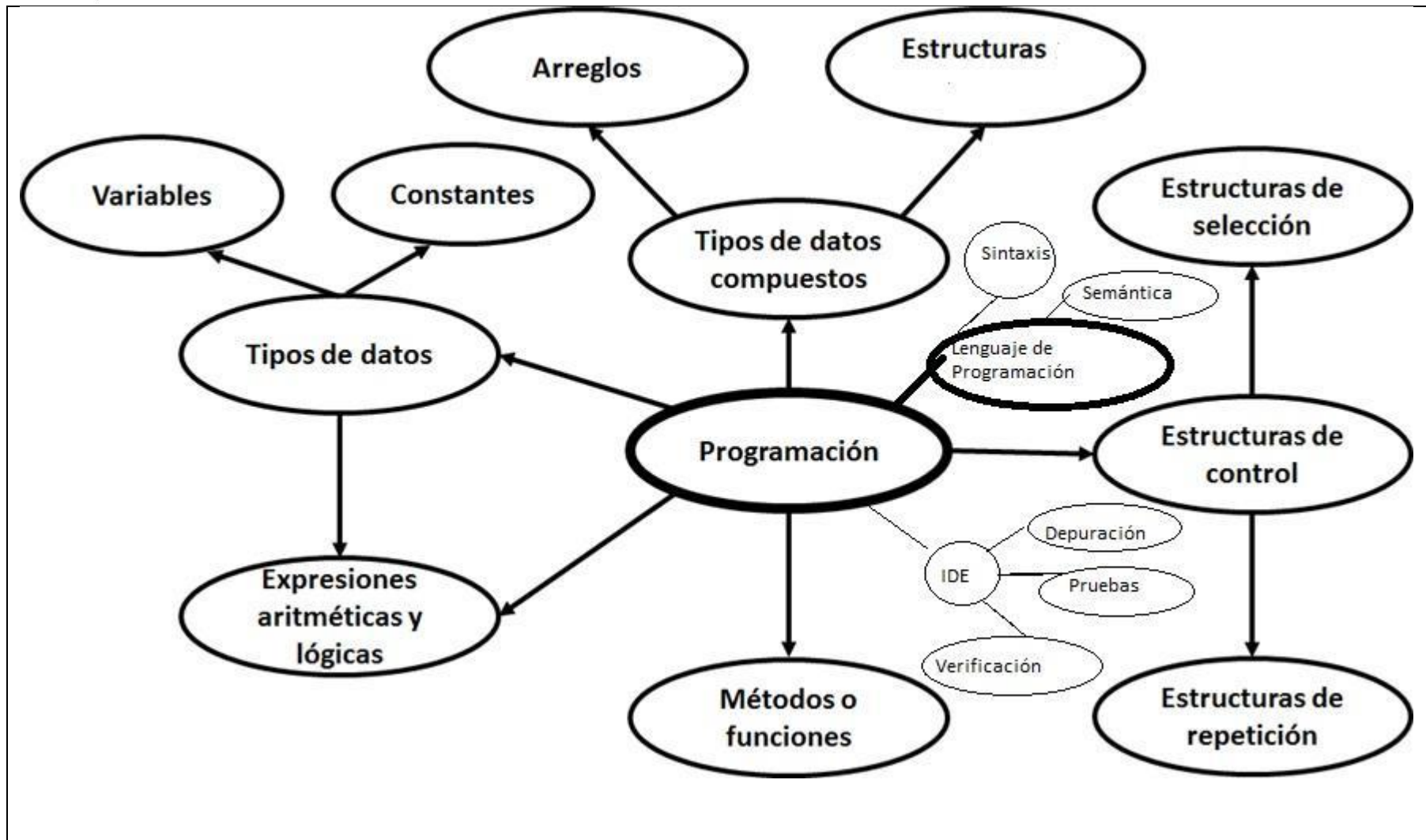
Competencias a desarrollar en la UA o Asignatura		
Transversales	Genéricas	Profesionales
<ul style="list-style-type: none"> ● Capacidad de abstracción, análisis y síntesis ● Identificar y resolver problemas ● Comprensión y construcción de procesos ● Capacidad de crítica y autocrítica ● Análisis de la realidad ● Toma de decisiones ● Capacidad de comunicación oral y escrita ● Motivar y conducir hacia metas comunes ● Trabajo en equipo y colaborativo ● Capacidad de aplicar conocimientos en la práctica ● Capacidad de organizar el tiempo ● Capacidad de actuar en nuevas situaciones ● Capacidad de aprender y actualizarse ● Trabajo autónomo ● Capacidad creativa ● Solidaridad ● Habilidades interpersonales 	<ul style="list-style-type: none"> ● Identifica de un problema dado los elementos necesarios para su solución utilizando un lenguaje de programación estructurado. ● Estructura la solución del problema en módulos individuales programables. ● Codifica, revisa, y ejecuta la solución del problema y subproblemas utilizando buenas prácticas de programación. 	<ul style="list-style-type: none"> ● Analiza, diseña, desarrolla e implementa soluciones relacionadas con hardware y software.
Saberes involucrados en la UA o Asignatura		
Saber (conocimientos)	Saber hacer (habilidades)	Saber ser (actitudes y valores)



UNIVERSIDAD DE GUADALAJARA

<ul style="list-style-type: none">● Elementos del paradigma de programación Estructurada.● Características de un compilador de Lenguaje Estructurado como herramienta para la codificación de la solución del problema● Lenguaje de programación Estructurado	<ul style="list-style-type: none">● Selecciona la solución de problemas utilizando la abstracción.● Usa un lenguaje de programación estructurado y un respectivo IDE.● Usa equipos de cómputo.● Integra los conceptos básicos de la Programación Estructurada en la solución de problemas	<ul style="list-style-type: none">● Respeto a la agenda propuesta para la materia: puntualidad y asistencia.● Entrega de trabajos originales.● Sentido crítico y análisis grupal a las soluciones individuales del grupo.● Fomento a la iniciativa entre los alumnos del grupo.
Producto Integrador Final de la UA o Asignatura		
<p>Título del Producto: Análisis, diseño y codificación de un sistema.</p> <p>Objetivo: Aplicar el paradigma de programación estructurada en un problema dado.</p> <p>Descripción: La solución al problema debe contener las siguientes características:</p> <ul style="list-style-type: none">● Análisis: identificar los elementos del problema y sus relaciones.● Diseño de la solución: mediante la utilización de diagramas y / o pseudocódigos.● Implementación: codificar considerando buenas prácticas de programación. La solución debe ser modular e incluir estructuras selectivas, repetitivas, datos estructurados como arreglos y registros.● Funcionalidad: el programa debe cumplir con las especificaciones establecidas.		

3. ORGANIZADOR GRÁFICO DE LOS CONTENIDOS DE LA UA O ASIGNATURA



4. SECUENCIA DEL CURSO POR UNIDADES TEMÁTICAS

Unidad temática 1: Conceptos básicos del Lenguaje C



Objetivo de la unidad temática: Aplicar los conceptos básicos de programación utilizando el entorno del compilador para la solución de un problema.

Introducción: En esta unidad se describirán los conceptos básicos del lenguaje de programación C. Mediante el empleo de entornos de desarrollo, el alumno editará, compilará y ejecutará sus primeros programas. Estos primeros programas servirán de base en unidades posteriores.

Contenido temático		Saberes involucrados	Producto de la unidad temática	
1.1 Historia del lenguaje C 1.2 Entorno del compilador 1.3 Elementos de un programa 1.3.1 Identificador 1.3.2 Tipos de datos primitivos 1.3.3 Definición y declaración de datos 1.3.3.1 Variables 1.3.3.2 Constantes (const, #define) 1.3.4 Operadores y expresiones 1.3.5 Palabras reservadas 1.3.6 Comentarios		Identifica y utiliza elementos básicos de programación tales como variables, constantes y operadores en procesos de entrada-salida, así como en expresiones para resolver problemas. Utiliza un lenguaje de programación para codificar las soluciones de los problemas. Identifica y organiza los elementos que se requiere para resolver un problema. Elabora y evalúa expresiones matemáticas simples, aplicando la correcta prioridad de los operadores. Presenta sus productos en tiempo y forma, de tal manera que demuestra interés y cuidado en su trabajo. Fortalece el trabajo colaborativo y en equipo.	Archivo .c con la codificación de por lo menos un programa que involucre el uso de variables, constantes y operadores. Donde exista la entrada, proceso y salida de datos. Bajo las siguientes características: <ul style="list-style-type: none"> ● Inicializar siempre las variables cuando se declaran. ● Expresar valores literales como constantes. 	
Actividades del docente	Actividades del estudiante	Evidencia de la actividad	Recursos materiales y	Tiempo destinado
Solicita a los estudiantes lecturas	Identificar las razones que llevaron a la	Reporte (archivo	Internet y / o	1



UNIVERSIDAD DE GUADALAJARA

previas de la historia del lenguaje C.	creación de este lenguaje, así como su evolución.	en Word o pdf)	bibliografía sugerida.	
Pide a los estudiantes la instalación de un entorno de desarrollo adecuado para el lenguaje C. Ejemplifica el uso del IDE para la codificación de un primer programa.	Aprender a utilizar las funciones básicas del entorno como crear archivo nuevo, guardar cambios, compilar y ejecutar.	Archivo .c de un primer programa	PC y el IDE (Entorno de Desarrollo Integrado)	2
El profesor indica a los alumnos organizarse en grupos para resolver una serie de ejercicios donde se apliquen los conceptos básicos. El profesor aclara las dudas de los grupos de trabajo.	Los alumnos discutirán la mejor manera para resolver los problemas. Los alumnos evaluarán expresiones aritméticas y lógicas presentadas por el profesor.	Código (archivos .c) de los programas funcionales que resuelven los problemas propuestos	PC y el IDE	5

Unidad temática 2: Programación Estructurada

Objetivo de la unidad temática: Crear programas utilizando el paradigma de Programación Estructurada.

Introducción: En esta unidad se explicarán las estructuras de control que maneja el lenguaje de programación C. Mediante el empleo del entorno de desarrollo instalado, el alumno desarrollará los programas que involucren estos temas. Estos programas servirán de base en unidades posteriores.

Contenido temático	Saberes involucrados	Producto de la unidad temática
2.1 Estructuras de control 2.1.1 Clasificación de estructuras selectivas 2.1.1.1 Secuencial 2.1.1.2 Selectiva simple (if) 2.1.1.3 Selectiva doble (if - else) 2.1.1.4 Selectiva múltiple (switch) 2.2 Estructuras de control repetitivas o de iteración condicional 2.2.1 Contadores,, acumuladores y banderas 2.2.2 Mientras (while)	Clasifica y aplica las diferentes estructuras de control en los problemas a resolver. Usa de manera correcta contadores, acumuladores y banderas. Usa de manera correcta los operadores lógicos y relacionales en las expresiones (condiciones).	Archivos .c de cada una de las estructuras de control, aplicando el uso de banderas, contadores y acumuladores. Utilizando buenas prácticas de programación como: <ul style="list-style-type: none"> ● Inicializar siempre las variables cuando se declaran.



UNIVERSIDAD DE GUADALAJARA

<p>2.2.3 Hacer mientras (do - while) 2.2.4 Desde (for) 2.3 Implementación de estructuras anidadas</p>	<p>Fortalece la abstracción en la solución de problemas.</p> <p>Mejora sus habilidades de comunicación oral y escrita.</p>	<ul style="list-style-type: none"> ● Expresar valores literales como constantes. ● Uso de sangría adecuado. ● Usar siempre llaves ({}) en las estructuras de control. ● Colocar al lado de una llave que cierre un bloque de código un indicativo de que tipo de estructura cierra. 		
Actividades del docente	Actividades del estudiante	Evidencia de la actividad	Recursos materiales y	Tiempo destinado
<p>Solicita a los estudiantes lecturas previas acerca de las estructuras selectivas: if, if - else, switch</p> <p>El profesor resuelve las dudas que tuvieron los estudiantes en la resolución de problemas con las sentencias selectivas.</p> <p>El profesor proporciona una serie adicional de ejercicios a resolver para fortalecer el uso de las estructuras selectivas.</p>	<p>Resolverá problemas en los que aplique sus conocimientos sobre las tres estructuras selectivas.</p>	<p>Códigos (archivos .c) de por lo menos un programa por cada una de las estructuras selectivas.</p>	<p>Internet y / o bibliografía sugerida.</p> <p>PC e IDE</p>	<p>10</p>
<p>Solicita a los estudiantes lecturas previas acerca de las estructuras repetitivas: while, do-while, for.</p> <p>El profesor resuelve las dudas que tuvieron los estudiantes en la</p>	<p>Resolverá problemas en los que aplique sus conocimientos sobre las estructuras repetitivas.</p>	<p>Códigos (archivos .c) de por lo menos un programa funcional por cada una de las</p>	<p>Internet y / o bibliografía sugerida.</p> <p>PC e IDE</p>	<p>10</p>



UNIVERSIDAD DE GUADALAJARA

<p>resolución de problemas con las estructuras repetitivas.</p> <p>El profesor proporciona una serie adicional de ejercicios a resolver para fortalecer el uso de las estructuras repetitivas.</p>		<p>estructuras.</p>		
<p>El profesor planteará una serie de ejercicios con mayor grado de complejidad a resolver con estructuras anidadas, tanto selectivas como repetitivas.</p> <p>Organiza un debate con las diferentes soluciones a la serie de ejercicios.</p>	<p>El alumno codificará las soluciones a una serie de ejercicios propuestos donde aplique las estructuras anidadas.</p> <p>Presenta y justifica sus propuestas de soluciones y discute con sus compañeros de grupo sobre las ventajas y desventajas de las diferentes soluciones con respecto a las demás presentadas en el grupo.</p>	<p>Códigos (archivos .c) de por lo menos un programa funcional por cada una de las estructuras. El archivo debe de estar debidamente comentado.</p>	<p>Internet y / o bibliografía sugerida.</p> <p>PC e IDE</p> <p>Cañón.</p>	<p>4</p>

Unidad temática 3: Arreglos

Objetivo de la unidad temática: Crear programas utilizando los diferentes tipos de arreglos de datos.

Introducción: En esta unidad se utilizarán los diferentes tipos de arreglos que maneja el lenguaje de programación C. Mediante el empleo del entorno de desarrollo instalado, el alumno desarrollará los programas que involucren estos temas. Estos programas servirán de base en unidades posteriores.

Contenido temático	Saberes involucrados	Producto de la unidad temática
<p>3.1 Definición</p> <p>3.2 Tipos de arreglos</p> <p>3.2.1 Vectores</p> <p>3.2.2 Matrices</p>	<p>Clasifica y aplica de manera adecuada los diferentes tipos de arreglos en los problemas a resolver.</p> <p>Presenta sus productos en tiempo y forma, de tal manera que demuestra interés y cuidado en su trabajo</p>	<p>Archivos .c por cada tipo de dato presentar un programa tanto en vectores como matrices (al menos 6 programas).</p> <p>Utilizando buenas prácticas de programación como:</p>



	<p>Refuerza la práctica del uso de las estructuras selectivas y repetitivas.</p> <p>Aplica operaciones sobre conjuntos de datos: ordena, analiza, promedia, entre otros.</p> <p>Aplica conocimientos en la práctica.</p> <p>Practica la toma de decisiones.</p>	<ul style="list-style-type: none"> ● Inicializar siempre las variables cuando se declaran. ● Expresar valores literales como constantes. ● Uso de sangría adecuado. ● Usar siempre llaves ({}) en las estructuras de control. ● Colocar al lado de una llave que cierre un bloque de código un indicativo de que tipo de estructura cierra.
--	---	---

Actividades del docente	Actividades del estudiante	Evidencia de la actividad	Recursos materiales y	Tiempo destinado
<p>Solicita a los estudiantes lecturas previas acerca de arreglos: vectores y matrices.</p> <p>El profesor planteará una serie de ejercicios con mayor grado de complejidad a resolver con arreglos, tanto vectores como matrices.</p> <p>Organiza un debate con las diferentes soluciones a la serie de ejercicios.</p>	<p>El alumno codificará las soluciones a una serie de ejercicios propuestos donde aplique los arreglos.</p> <p>Presenta y justifica sus propuestas de soluciones y discute con sus compañeros de grupo sobre las ventajas y desventajas de las diferentes soluciones con respecto a las demás presentadas en el grupo</p>	<p>Códigos (archivos .c) de por lo menos un programa funcional por cada uno de los tipos de arreglos. El archivo debe de estar debidamente comentado.</p>	<p>Internet y / o bibliografía sugerida.</p> <p>PC e IDE</p> <p>Cañón.</p>	<p>8</p>

Unidad temática 4: Manejo de funciones

Objetivo de la unidad temática: Crear programas aplicando la programación modular.

Introducción: En esta unidad se aplicará el principio de “divide y vencerás” en la solución de problemas utilizando los distintos tipos de funciones que maneja el lenguaje de programación C. Mediante el empleo del entorno de desarrollo instalado, el alumno desarrollará los



programas que involucren estos temas. Estos programas servirán de base en unidades posteriores

Contenido temático	Saberes involucrados	Producto de la unidad temática
<p>4.1 Definición</p> <p>4.2 Tipos de funciones</p> <p>4.2.1 Funciones sin paso de parámetros</p> <p>4.2.2 Funciones con paso de parámetros</p> <p>4.2.2.1 Funciones con parámetros (tipos de datos primitivos)</p> <p>4.2.2.2 Funciones con parámetros (tipo arreglo)</p> <p>4.2.3 Funciones predefinidas</p> <p>4.2.3.1 Matemáticas</p> <p>4.2.3.2 De manejo de caracteres</p>	<p>Analiza y diseña una solución descendente de un problema dado. Descompone el problema en subproblemas.</p> <p>Codifica los subproblemas como funciones.</p> <p>Decide qué tipo de función implementa de acuerdo a cada subproblema.</p> <p>Usa adecuadamente variables locales y globales en la solución de sus problemas.</p> <p>Presenta sus productos en tiempo y forma, de tal manera que demuestra interés y cuidado en su trabajo</p>	<p>Codificación de programas modulares que resuelvan problemas sencillos y que involucren el uso de los distintos tipos de funciones.</p> <p>Archivos .c por cada tipo de dato presentar un programa tanto en vectores como matrices (al menos 6 programas).</p> <p>Utilizando buenas prácticas de programación como:</p> <ul style="list-style-type: none">● Inicializar siempre las variables cuando se declaran.● Expresar valores literales como constantes.● Usar de manera adecuada las variables globales y locales.● Evitar el uso de sentencias goto, break.● Usar un único return por función, que se colocará como última sentencia de la función.



			<ul style="list-style-type: none"> ● Evitar escribir funciones y procedimientos demasiado largos. ● Evitar copiar y pegar segmentos casi idénticos de código un mismo programa. ● Usar siempre llaves ({}) en las estructuras de control. ● Colocar al lado de una llave que cierre un bloque de código un indicativo de que tipo de estructura cierra. 	
Actividades del docente	Actividades del estudiante	Evidencia de la actividad	Recursos materiales y	Tiempo destinado
<p>Solicita a los estudiantes lecturas previas acerca de funciones en C.</p> <p>El profesor planteará una serie de ejercicios con mayor grado de complejidad a resolver con todos los tipos de funciones.</p> <p>Organiza un debate con las diferentes soluciones a la serie de ejercicios.</p>	<p>El alumno codificará las soluciones a una serie de ejercicios propuestos donde aplique los diferentes tipos de funciones.</p> <p>Presenta y justifica sus propuestas de soluciones y discute con sus compañeros de grupo sobre las ventajas y desventajas de las diferentes soluciones con respecto a las demás presentadas en el grupo</p>	<p>Códigos (archivos .c) de por lo menos un programa funcional por cada uno de los tipos de funciones. El archivo debe de estar debidamente comentado.</p>	<p>Internet y / o bibliografía sugerida.</p> <p>PC e IDE</p> <p>Cañón.</p>	<p>16</p>
Unidad temática 5: Registros				
<p>Objetivo de la unidad temática: Crear programas en donde se requiera el diseño de datos compuestos complejos a partir de los datos simples y arreglos de datos</p>				



Introducción: En esta unidad se aplicará el uso de registros para manipular la información de una forma organizada. Mediante el empleo del entorno de desarrollo instalado, el alumno desarrollará los programas que involucren estos temas. Estos programas servirán de base para UA posteriores.

Contenido temático	Saberes involucrados	Producto de la unidad temática
<p>5.1 Definición</p> <p>5.2 Operaciones con registros</p> <p>5.2.1 Entrada de datos de los elementos de un registro.</p> <p>5.2.2 Salida de datos de un registro</p> <p>5.3 Arreglos de registros</p> <p>5.3.1 Definición</p> <p>5.3.2 Operaciones con arreglos de registros</p> <p>5.3.2.1 Entrada de datos de los elementos de un arreglo de registros</p> <p>5.3.2.2 Salida de datos de un arreglo de registros.</p>	<p>Abstrae el problema organizando la información en tipos de datos compuestos complejos.</p> <p>Presenta sus productos en tiempo y forma, de tal manera que demuestra interés y cuidado en su trabajo</p> <p>Refuerza la práctica del uso de las estructuras selectivas y repetitivas.</p> <p>Aplica operaciones sobre conjuntos de datos: ordena, analiza, promedia, entre otros.</p> <p>Aplica conocimientos en la práctica.</p> <p>Practica la toma de decisiones.</p>	<p>Codificación de programas modulares que resuelvan problemas sencillos y que involucren el uso de registros de datos.</p> <p>Archivos .c con registros (al menos 1 programa).</p> <p>Utilizando buenas prácticas de programación como:</p> <ul style="list-style-type: none">● Inicializar siempre las variables cuando se declaran.● Expresar valores literales como constantes.● Usar de manera adecuada las variables globales y locales.● Evitar el uso de sentencias goto, break.● Usar un único return por función, que se colocará como última sentencia de la función.● Evitar escribir funciones y



UNIVERSIDAD DE GUADALAJARA

		<p>procedimientos demasiado largos.</p> <ul style="list-style-type: none"> ● Evitar copiar y pegar segmentos casi idénticos de código un mismo programa. ● Usar siempre llaves ({}) en las estructuras de control. ● Colocar al lado de una llave que cierre un bloque de código un indicativo de que tipo de estructura cierra. 		
Actividades del docente	Actividad del estudiante	Evidencia de la actividad	Recursos materiales y	Tiempo destinado
<p>Solicita a los estudiantes lecturas previas acerca de registros.</p> <p>El profesor planteará por lo menos un ejercicio a resolver con registros en el cual se apliquen todas las unidades temáticas previas.</p> <p>Organiza una exposición de las diferentes soluciones que hubieran realizado los alumnos.</p>	<p>El alumno codificará la solución al ejercicio propuesto donde aplique registros.</p> <p>Expone y justifica su propuesta de solución, discute con sus compañeros de grupo sobre las ventajas y desventajas de la solución con respecto a las demás presentadas en el grupo</p>	<p>Código (archivo .c) de por lo menos un programa funcional que incluya el uso de registros. El archivo debe de estar debidamente comentado.</p>	<p>Internet y / o bibliografía sugerida.</p> <p>PC e IDE</p> <p>Cañón.</p>	<p>12</p>



5. EVALUACIÓN Y CALIFICACIÓN

Requerimientos de acreditación:

De acuerdo al “REGLAMENTO GENERAL DE EVALUACIÓN Y PROMOCIÓN DE ALUMNOS DE LA UNIVERSIDAD DE GUADALAJARA”:

Artículo 5. “El resultado final de las evaluaciones será expresado conforme a la escala de calificaciones centesimal de 0 a 100, en números enteros, considerando como mínima aprobatoria la calificación de 60.”

Artículo 20. “Para que el alumno tenga derecho al registro del resultado final de la evaluación en el periodo ordinario, establecido en el calendario escolar aprobado por el H. Consejo General Universitario, se requiere:

- I. Estar inscrito en el plan de estudios y curso correspondiente, y

Tener un mínimo de asistencia del 80% a clases y actividades registradas durante el curso.”

De acuerdo al “REGLAMENTO GENERAL DE EVALUACIÓN Y PROMOCIÓN DE ALUMNOS DE LA UNIVERSIDAD DE GUADALAJARA”:

Artículo 27. “Para que el alumno tenga derecho al registro de la calificación en el periodo extraordinario, se requiere:

- I. Estar inscrito en el plan de estudios y curso correspondiente.
- II. Haber pagado el arancel y presentar el comprobante correspondiente.
- III. Tener un mínimo de asistencia del 65% a clases y actividades registradas durante el curso.”

Criterios generales de evaluación:

Esta UA requiere de la presentación de programas en Lenguaje C, los cuales deben cumplir con lo siguiente:

- Entrega en tiempo
- Los códigos deben estar comentados con datos de quien lo elabora, la fecha de elaboración, descripción del programa.
- Queda estrictamente prohibido el plagio

Para la exposición el alumno debe mostrar plena comprensión del contenido de su proyecto y responder satisfactoriamente a los cuestionamientos orales que se le hagan del mismo.



Evidencias o Productos			
Evidencia o producto	Competencias y saberes involucrados	Contenidos temáticos	Ponderación
<p>Archivo .c con la codificación de por lo menos un programa que involucre el uso de variables, constantes y operadores. Donde exista la entrada, proceso y salida de datos.</p> <p>Bajo las siguientes características:</p> <ul style="list-style-type: none"> ● Inicializar siempre las variables cuando se declaran. ● Expresar valores literales como constantes. 	<p>Identifica y utiliza elementos básicos de programación tales como variables, constantes y operadores en procesos de entrada-salida, así como en expresiones para resolver problemas.</p> <p>Utiliza un lenguaje de programación para codificar las soluciones de los problemas.</p> <p>Identifica y organiza los elementos que se requiere para resolver un problema.</p> <p>Elabora y evalúa expresiones matemáticas simples, aplicando la correcta prioridad de los operadores.</p> <p>Presenta sus productos en tiempo y forma, de tal manera que demuestra interés y cuidado en su trabajo.</p> <p>Fortalece el trabajo colaborativo y en equipo.</p>	<p>Historia del lenguaje C</p> <p>Entorno del IDE</p> <p>Elementos del lenguaje C</p> <p>Identificador</p> <p>Tipos de datos primitivos</p> <p>Variables</p> <p>Constantes</p> <p>Operadores y expresiones</p> <p>Palabras reservadas</p> <p>Comentarios</p>	<p>5%</p>
<p>2.</p> <p>Archivos .c de cada una de las estructuras de control, aplicando el uso de banderas, contadores y acumuladores.</p>	<p>Clasifica y aplica las diferentes estructuras de control en los problemas a resolver.</p> <p>Usa de manera correcta contadores, acumuladores y banderas.</p> <p>Usa de manera correcta los operadores</p>	<p>Estructuras selectivas:</p> <p>Simple (if)</p> <p>Doble (if - else)</p> <p>Múltiple (switch)</p> <p>Estructuras repetitivas:</p> <p>Contadores</p>	<p>15%</p>



UNIVERSIDAD DE GUADALAJARA

<p>Utilizando buenas prácticas de programación como:</p> <ul style="list-style-type: none">● Inicializar siempre las variables cuando se declaran.● Expresar valores literales como constantes.● Uso de sangría adecuado.● Usar siempre llaves ({}) en las estructuras de control.● Colocar al lado de una llave que cierre un bloque de código un indicativo de que tipo de estructura cierra.	<p>lógicos y relacionales en las expresiones (condiciones).</p> <p>Fortalece la abstracción en la solución de problemas.</p> <p>Mejora sus habilidades de comunicación oral y escrita.</p>	<p>Acumuladores Banderas Mientras (while) Hacer mientras (do - while) Desde (for)</p> <p>Estructuras anidadadas</p>	
<p>3.</p> <p>Archivos .c por cada tipo de dato presentar un programa tanto en vectores como matrices (al menos 6 programas).</p> <p>Utilizando buenas prácticas de programación como:</p> <ul style="list-style-type: none">● Inicializar siempre las variables cuando se declaran.● Expresar valores literales como constantes.● Uso de sangría adecuado.● Usar siempre llaves ({}) en las estructuras de control.● Colocar al lado de una llave que cierre un bloque de código un indicativo de que	<p>Clasifica y aplica de manera adecuada los diferentes tipos de arreglos en los problemas a resolver.</p> <p>Presenta sus productos en tiempo y forma, de tal manera que demuestra interés y cuidado en su trabajo</p> <p>Refuerza la práctica del uso de las estructuras selectivas y repetitivas.</p> <p>Aplica operaciones sobre conjuntos de datos: ordena, analiza, promedia, entre otros.</p> <p>Aplica conocimientos en la práctica.</p> <p>Practica la toma de decisiones.</p>	<p>Definición de arreglo.</p> <p>Tipos de arreglos: Vectores Matrices</p>	<p>20%</p>



tipo de estructura cierra.			
<p>4.</p> <p>Codificación de programas modulares que resuelvan problemas sencillos y que involucren el uso de los distintos tipos de funciones.</p> <p>Archivos .c por cada tipo de dato presentar un programa tanto en vectores como matrices (al menos 6 programas).</p> <p>Utilizando buenas prácticas de programación como:</p> <ul style="list-style-type: none">● Inicializar siempre las variables cuando se declaran.● Expresar valores literales como constantes.● Usar de manera adecuada las variables globales y locales.● Evitar el uso de sentencias goto, break.● Usar un único return por función, que se colocará como última sentencia de la función.● Evitar escribir funciones y procedimientos demasiado largos.● Evitar copiar y pegar segmentos casi idénticos	<p>Analiza y diseña una solución descendente de un problema dado. Descompone el problema en subproblemas.</p> <p>Codifica los subproblemas como funciones.</p> <p>Decide qué tipo de función implementa de acuerdo a cada subproblema.</p> <p>Usa adecuadamente variables locales y globales en la solución de sus problemas.</p> <p>Presenta sus productos en tiempo y forma, de tal manera que demuestra interés y cuidado en su trabajo</p>	<p>Definición de función</p> <p>Tipos de funciones:</p> <p>Sin paso de parámetros</p> <p>Con paso de parámetros:</p> <ul style="list-style-type: none">*Datos primitivos*Arreglo <p>Funciones predefinidas:</p> <p>Matemáticas</p> <p>De manejo de caracteres</p>	<p>15 %</p>



UNIVERSIDAD DE GUADALAJARA

<p>de código un mismo programa.</p> <ul style="list-style-type: none">● Usar siempre llaves ({}) en las estructuras de control.● Colocar al lado de una llave que cierre un bloque de código un indicativo de que tipo de estructura cierra.			
<p>5.</p> <p>Codificación de programas modulares que resuelvan problemas sencillos y que involucren el uso de registros de datos.</p> <p>Archivos .c con registros (al menos 1 programa).</p> <p>Utilizando buenas prácticas de programación como:</p> <ul style="list-style-type: none">● Inicializar siempre las variables cuando se declaran.● Expresar valores literales como constantes.● Usar de manera adecuada las variables globales y locales.● Evitar el uso de sentencias goto, break.● Usar un único return por función, que se colocará	<p>Abstrae el problema organizando la información en tipos de datos compuestos complejos.</p> <p>Presenta sus productos en tiempo y forma, de tal manera que demuestra interés y cuidado en su trabajo</p> <p>Refuerza la práctica del uso de las estructuras selectivas y repetitivas.</p> <p>Aplica operaciones sobre conjuntos de datos: ordena, analiza, promedia, entre otros.</p> <p>Aplica conocimientos en la práctica.</p> <p>Practica la toma de decisiones.</p>	<p>Definición de registro</p> <p>Operaciones con registros: Entrada y salida de datos en registros.</p> <p>Arreglos de registros: Entrada, salida y manejo de datos en arreglos de registros.</p>	<p>15%</p>



UNIVERSIDAD DE GUADALAJARA

<p>como última sentencia de la función.</p> <ul style="list-style-type: none"> ● Evitar escribir funciones y procedimientos demasiado largos. ● Evitar copiar y pegar segmentos casi idénticos de código un mismo programa. ● Usar siempre llaves ({}) en las estructuras de control. ● Colocar al lado de una llave que cierre un bloque de código un indicativo de que tipo de estructura cierra. 			
Producto final			
Descripción		Evaluación	
<p>Título: Análisis, diseño y codificación de un sistema de cómputo.</p>	<p>Criterios de fondo: Sistema funcional en base a los requerimientos establecidos.</p> <p>Criterios de forma: Presenta en tiempo y forma el sistema. Aplica las buenas prácticas de programación.</p>	<p>Ponderación</p>	
<p>Objetivo: Aplicar el paradigma de programación estructurada en un problema dado.</p>		<p>25%</p>	
<p>Caracterización La solución al problema debe contener las siguientes características:</p> <ul style="list-style-type: none"> ● Análisis: identificar los elementos del problema y sus relaciones. ● Diseño de la solución: mediante la utilización de diagramas y / o pseudocódigos. ● Implementación: codificar considerando buenas prácticas de programación. La solución debe ser modular e incluir estructuras selectivas, repetitivas, datos estructurados como arreglos y registros. ● Funcionalidad: el programa debe cumplir con las especificaciones establecidas. 			



UNIVERSIDAD DE GUADALAJARA

Otros criterios		
Criterio	Descripción	Ponderación
Exposición del proyecto.	Expone y justifica su propuesta de solución, discute con sus compañeros de grupo sobre las ventajas y desventajas de la solución con respecto a las demás presentadas en el grupo	5%



6. REFERENCIAS Y APOYOS

Referencias bibliográficas

Referencias básicas

Autor (Apellido, Nombre)	Año	Título	Editorial	Enlace o biblioteca virtual donde esté disponible (en su caso)
Joyanes Aguilar, L	2020	Fundamentos de programación	McGraw Hill. 5ta. Edición	

Referencias complementarias

Márquez G., Osorio S., Olvera N	2011	Introducción a la Programación Estructurada en C	Pearson	
Juganaru Mathieu, M	2014	Introducción a la Programación	Grupo Editorial Patria	https://www.academia.edu/31289773/Introducci%C3%B3n_a_la_programaci%C3%B3n
Sznajdleder, Pablo Augusto	2017	PROGRAMACIÓN ESTRUCTURADA A FONDO Implementación de algoritmos en C	Alfaomega	