



1.- Identificación de la Unidad de Aprendizaje					
Nombre de la Unidad de Aprendizaje					
<i>Estructuras de Datos I</i>					
Clave de la UA	Modalidad de la UA	Tipo de UA		Valor de créditos	Área de formación
IH060	Presencial	Curso-taller		8	Básica Particular Obligatoria
Hora semana		Horas teoría/semestre	Horas práctica/semestre	Total de horas:	Seriación
4		60	20	80	Algoritmia
Departamento			Academia		
Ciencias de la Información y Desarrollos Tecnológicos			Ingeniería de Software		
Presentación					
<p>El alumno aplicará el paradigma de programación Orientada a Objetos elaborando programas, enfatizando en el análisis descendente de la solución a través del diseño de algoritmos en los lenguajes Java y C++. Para ello dispondrá de los fundamentos teóricos básicos que lo constituyen. Utilizará estructuras de Datos Lineales (Listas, pilas, colas) y no-lineales (árboles y grafos) basándose en técnicas de implementación de algoritmos de búsqueda y ordenación.</p>					
Tipos de saberes					
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)			
<p>Conocer los elementos que constituyen los entornos de programación.</p> <p>Reconocer el ámbito o alcance de los tipos de datos primitivos y sus operadores.</p> <p>Control del flujo de la ejecución.</p> <p>Control de excepciones o rupturas durante el tiempo de ejecución en</p>	<p>Abstrae y sintetiza escenarios o problemáticas de la realidad y sus posibles soluciones.</p> <p>Aplica los conocimientos teóricos en la práctica.</p> <p>capacidad de comunicación oral y escrita.</p> <p>Capacidad de integración a grupos de trabajo.</p> <p>Autónomo en el aprendizaje.</p>	<p>Receptivo y empático al nuevo conocimiento, ya sean nuevos paradigmas de programación y/o nuevas tecnologías.</p> <p>Ser respetuoso a las opiniones y trabajo de sus compañeros.</p> <p>Ser creativo.</p> <p>Ser responsable.</p> <p>Ser colaborativo.</p> <p>Ser proactivo.</p> <p>Ser honesto</p> <p>Ser organizado</p>			

**Comentado [1]:** Se describe al final

**Comentado [2]:** Se describe al final ya que es la recopilación de los conocimientos, habilidades, actitudes y valores de cada uno de los módulos descritos mas adelante



implementaciones con punteros.  Diseñar y construir soluciones informáticas utilizando tipos de dato abstracto (TDA).  Pertinencia y Teoría de Objetos  Implementa soluciones explotando los tipos de dato abstracto con las ventajas del contexto orientado a objetos.  Conocer e identificar la estructura, organización y funcionamiento de los Tipos de dato Abstracto no lineales.  Compara alternativas de implementación de las estructuras de datos.  Saber implementar las distintas estructuras de datos y sus operaciones con plantillas.	Capacidad de síntesis y análisis.  Trabaja colaborativamente.  Capacidad de análisis y síntesis.  Utilizar eficientemente los recursos y herramientas disponibles del lenguaje seleccionado para la materia.  Hábil en el razonamiento crítico.	Ser analítico
<b>Competencia genérica</b>		<b>Competencia profesional</b>
Capacidad de análisis, síntesis y crítica al implementar modelos abstractos en la resolución de problemas.	Resuelve eficientemente problemas de manipulación y almacenamiento de datos seleccionando el tipo de dato abstracto (TDA) más conveniente.	
<b>Saberes previos del alumno</b>		
El alumno deberá tener conocimiento en: Lógica de programación, estructuras del control de flujo, programación modular o por funciones, teoría de objetos, identificar y sintetizar escenarios o problemáticas, saber trabajo en equipo.		
<b>Perfil de egreso al que se abona</b>		

**Comentado [3]:** Esto puede redactarse al iniciar el desarrollo del programa

**Comentado [4]:** Esto se hace hasta el final ya es la recopilación de las competencias específicas de cada módulo

**Comentado [5]:** Esto si se puede realizar al inicio del desarrollo del programa

**Comentado [6]:** Podría ser algo así?

"El alumno deberá de contar con los conocimientos adquiridos en las materias de fundamentos de programación y programación orientada a objetos."

Manejo de los lenguajes de programación C++ y JAVA

**Comentado [7]:** Esto si se puede realizar al inicio del desarrollo del programa



Desarrollo de sistemas de información, capacidad de análisis y diseño de soluciones alternativas e innovadoras a problemas utilizando diversas herramientas y metodologías.

#### Perfil deseable del docente

##### Formación profesional.

El docente deberá contar preferentemente con un posgrado y/o al menos con una licenciatura afín al área de cómputo, deberá tener amplia experiencia en desarrollo de software orientado a objetos, además de actualizarse periódicamente en las versiones del lenguaje utilizado en la materia.

##### Habilidades.

Experiencia en:

Elaboración de planeaciones didácticas,

Diseño y evaluación de estrategias y actividades de aprendizaje,

Uso de tecnologías de la información y la comunicación.

Facilidad de palabra

**Comentado [8]:** Esto si se puede realizar al inicio del desarrollo del programa

## 2.- Contenidos temáticos

### Contenido

1. PROGRAMACIÓN ORIENTADA A OBJETOS EN C++
  - 1.1 Encapsulamiento
  - 1.2 Estructuras vs.clases
  - 1.3 Constructores y destructores
  - 1.4 Sobrecarga
  - 1.5 Herencia
  - 1.6 Polimorfismo
2. ESTRUCTURAS DE DATOS LINEALES
  - 2.1 Arreglos
  - 2.2 Pilas
  - 2.3 Colas
  - 2.4 Listas
  - 2.5 Listas Doblemente Ligadas
  - 2.6 Vectores
3. ESTRUCTURAS DE DATOS NO-LINEALES
  - 3.1 Árboles
  - 3.2 Grafos
4. ESTRUCTURAS DE DATOS PARAMÉTRICAS
  - 4.1 Templates o plantillas
  - 4.2 Arreglos con plantillas

**Comentado [9]:** Todo este apartado se puede realizar de forma consecutiva; hay que desarrollar cada uno de los módulos. Cuando se tenga la información de los apartados de competencias específicas y tipos de saberes, se podrá utilizar en el apartado 1



4.3 Estructuras lineales con plantillas		
4.4 Estructuras no-lineales con plantillas		
<b>Estrategias generales para impartir la unidad de aprendizaje</b>		
Solución de problemas y usos de los tipos de datos abstractos en aplicaciones de software.		
<b>Módulo I</b>		
1. PROGRAMACIÓN ORIENTADA A OBJETOS EN C++		
1.1 Encapsulamiento		
1.2 Estructuras vs.clases		
1.3 Constructores y destructores		
1.4 Sobrecarga		
1.5 Herencia		
1.6 Polimorfismo		
<b>Competencia Específica</b>		
Aplica fundamentos matemáticos, principios algorítmicos y teorías de Ciencias de la computación para simplificar el proceso en la modelación y diseño de soluciones informáticas, manejando eficientemente el espacio de memoria y minimizando el tiempo de acceso a los datos .		
<b>Tipos de saberes</b>		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Conocer los elementos que constituyen los entornos de programación.	Abstrae y sintetiza escenarios o problemáticas de la realidad.	Receptivo y empático al nuevo conocimiento, ya sean nuevos paradigmas de programación y/o nuevas tecnologías.
Reconocer el ámbito o alcance de los tipos de datos primitivos y sus operadores.	Aplica los conocimientos teóricos en la práctica.	Ser respetuoso a las opiniones y trabajo de sus compañeros.
Control del flujo de la ejecución.	capacidad de comunicación oral y escrita.	Ser organizado.
Control de excepciones o rupturas durante el tiempo de ejecución en implementaciones con punteros.	Capacidad de integración a grupos de trabajo.	
Diseñar y construir soluciones informáticas utilizando tipos de dato abstracto (TDA).	Autónomo en el aprendizaje.	
	Abstrae problemáticas y posibles soluciones.	
	Trabaja colaborativamente.	



Pertinencia y Teoría de Objetos		
<b>Módulo II</b>		
2. ESTRUCTURAS DE DATOS LINEALES		
2.1 Arreglos		
2.2 Pilas		
2.3 Colas		
2.4 Listas		
2.5 Listas Doblemente Ligadas		
2.6 Vectores		
<b>Competencia Específica</b>		
Aplica fundamentos matemáticos, principios algorítmicos y el paradigma orientado a objetos en el proceso de modelado y diseño de Tipos de Dato Abstracto para la soluciones a problemáticas de la vida real.		
<b>Tipos de saberes</b>		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Implementa soluciones explotando los tipos de dato abstracto con las ventajas del contexto orientado a objetos.	Capacidad de síntesis y análisis. Aplicar la teoría en la práctica. Abstrae y sintetiza escenarios o problemáticas de la realidad.	Ser creativo. Ser responsable. Ser colaborativo. Ser proactivo. Ser respetuoso Ser honesto Ser organizado
<b>Módulo III</b>		
3. ESTRUCTURAS DE DATOS NO-LINEALES		
3.1 Árboles		
3.2 Grafos		
<b>Competencia Específica</b>		
Comprende y organiza datos en memoria principal al aplicar criterios cualitativos y cuantitativos en el diseño de estrategias y escenarios donde las estructuras de datos no lineales son la óptima solución.		
<b>Tipos de saberes</b>		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Conocer e identificar la estructura, organización y funcionamiento de los Tipos de dato Abstracto no lineales.	Abstrae problemáticas y posibles soluciones. Aplica la teoría en la práctica. Trabaja colaborativamente.	Ser analítico. Ser responsable. Ser organizado. Ser disciplinado

**Comentado [10]:** Estructuras de Datos Lineales



Compara alternativas de implementación de las estructuras de datos.		
<b>Módulo IV</b>		
4. ESTRUCTURAS DE DATOS PARAMÉTRICAS		
4.1 Templates o plantillas		
4.2 Arreglos con plantillas		
4.3 Estructuras lineales con plantillas		
4.4 Estructuras no-lineales con plantillas		
<b>Competencia Específica</b>		
Identifica, define y aplica de forma eficiente plantillas de TDA's (modelos y sus operaciones) a la resolución de problemas.		
<b>Tipos de saberes</b>		
Saber (Conocimientos)	Saber hacer (Habilidades)	Saber ser (Actitudes y valores)
Saber implementar las distintas estructuras de datos y sus operaciones con plantillas.	capacidad de análisis y síntesis. Utilizar eficientemente los recursos y herramientas disponibles del lenguaje seleccionado para la materia. hábil en el razonamiento crítico.	Ser receptivo y empático al nuevo conocimiento, ya sean nuevos paradigmas de programación y/o nuevas tecnologías.  Ser analítico.  Ser responsable.  Ser organizado. Ser disciplinado
<b>Bibliografía básica</b>		
<ul style="list-style-type: none"> <li>• Weiss, M (2004). Data Structures and Algorithm Analysis in C++. Cuarta edición. Editorial Pearson Education.</li> <li>• Goodrich, M. (2011) Data Structures and Algorithms in C++. 2a edición, Editorial Wiley and Sons.</li> <li>• Deitel, Deitel, et. al. (2010) C++ How to Program. Editorial Prentice Hall.</li> </ul>		
<b>Bibliografía complementaria</b>		
<ul style="list-style-type: none"> <li>• Algorithms and data structures in C/C++, recurso en línea disponible en: <a href="https://www.cprogramming.com/algorithms-and-data-structures.html">https://www.cprogramming.com/algorithms-and-data-structures.html</a></li> </ul>		
<b>3.-Evaluación</b>		
<b>Criterios de Evaluación (% por criterio)</b>		

**Comentado [11]:** Se especifica después de haber descritos cada módulo

**Comentado [12]:** Se realiza después haber descrito el punto 2



Evaluación diagnóstica	
<ul style="list-style-type: none"><li>● Examen de programación básica en C</li><li>● Examen de programación básica en Java</li></ul>	
Evaluación Formativa	
<ul style="list-style-type: none"><li>● Resolución de problemas en el pizarrón</li><li>● Resolución de problemas en programas</li><li>● Exámenes rápidos</li></ul>	
Evaluación Sumativa	
<ul style="list-style-type: none"><li>● Exámenes teóricos escritos o mediante plataforma</li></ul>	45%
<ul style="list-style-type: none"><li>● Entrega de prácticas, programas y tareas</li></ul>	50%
<ul style="list-style-type: none"><li>● Autoevaluación</li></ul>	5%
<b>4.-Acreditación</b>	
De acuerdo al <b>REGLAMENTO GENERAL DE EVALUACIÓN Y PROMOCIÓN DE ALUMNOS DE LA UNIVERSIDAD DE GUADALAJARA</b> que señala:	
<p><b>Artículo 5.</b> 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. Las materias que no son sujetas a medición cuantitativa, se certificarán como acreditadas (A) o no acreditadas (NA).</p>	
<p><b>Artículo 20.</b> 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 II. Tener un mínimo de asistencia del 80% a clases y actividades registradas durante el curso.</p>	
<p><b>Artículo 25.</b> La evaluación en periodo extraordinario se calificará atendiendo a los siguientes criterios: <b>I.</b> La calificación obtenida en periodo extraordinario, tendrá una ponderación del 80% para la calificación final; <b>II.</b> La calificación obtenida por el alumno durante el periodo ordinario, tendrá una ponderación del 40% para la calificación en periodo extraordinario, y <b>III.</b> La calificación final para la evaluación en periodo extraordinario será la que resulte de la suma de los puntos obtenidos en las fracciones anteriores.</p>	
<p><b>Artículo 27.</b> 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.</p>	
<p><b>Artículo 33.</b> El alumno que por cualquier circunstancia no logre una calificación aprobatoria en el periodo extraordinario, deberá repetir la materia en el ciclo escolar inmediato siguiente en que se ofrezca, teniendo la oportunidad de acreditarla durante el proceso de evaluación ordinario o en el periodo extraordinario, excepto para alumnos de posgrado. En caso de que el alumno no logre acreditar la materia en los términos de este artículo, será dado de baja.</p>	



<b>5.- Participantes en la elaboración</b>	
Código	Nombre
	Cárdenas Larios Rigoberto
	Corona Pérez Manuel
2944934	García García César Alejandro
9814671	Mireles Mercado Marisela
	Patíño Carlos
<b>6.- Fecha de elaboración</b>	
<b>Diciembre 2019</b>	