

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

COORDINACIÓN GENERAL DE FORMACIÓN PROFESIONAL

PROGRAMA DE UNIDAD DE APRENDIZAJE

I. DATOS DE IDENTIFICACIÓN

- 1. Unidad Académica:** Facultad de Ingeniería, Arquitectura y Diseño, Ensenada; Facultad Ciencias Químicas e Ingeniería, Tijuana; y Facultad de Ciencias de la Ingeniería y Tecnología, Valle de las Palmas
- 2. Programa Educativo:** Ingeniero en Software y Tecnologías Emergentes
- 3. Plan de Estudios:** 2022-1
- 4. Nombre de la Unidad de Aprendizaje:** Estructuras de Datos
- 5. Clave:** 40007
- 6. HC:** 02HT: 01HL: 02HPC: 00HCL: 00HE: 02CR: 07
- 7. Etapa de Formación a la que Pertenece:** Disciplinaria
- 8. Carácter de la Unidad de Aprendizaje:** Obligatoria
- 9. Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



Equipo de diseño de PUA

David Abdel Mejía Medina.
Elitania Jiménez García
Thelma Violeta Ocegueda Miramontes
Rodrigo Lara Melgoza

Firma

Vo.Bo. de subdirector(es) de Unidad(es) Académica(s)

Humberto Cervantes De Ávila
Daniela Mercedes Martínez Platas
Noemí Hernández Hernández

Firma

Fecha: 23 de febrero de 2021

II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

La unidad de aprendizaje tiene como finalidad que el alumno conozca las estructuras de datos para que sea capaz de seleccionar y aplicar las mismas de manera adecuada en la construcción de software de procesamiento de información de forma eficiente, permitiéndole al alumno desarrollar habilidades de análisis y programación.

Esta unidad de aprendizaje forma parte de la etapa disciplinaria, es de carácter obligatoria, pertenece al área de conocimiento de métodos y tecnologías de software de la licenciatura de Ingeniero en Software y Tecnologías Emergentes y no es necesario haber aprobado una unidad de aprendizaje previa para cursarla.

III. COMPETENCIA GENERAL DE LA UNIDAD DE APRENDIZAJE

Implementar programas de computadora eficientes, a través de la selección de las estructuras de datos más adecuadas, para solucionar problemas de procesamiento de información, de manera creativa, analítica y proactiva.

IV. EVIDENCIA(S) DE APRENDIZAJE

Portafolio de evidencias de reportes de prácticas que incluya descripciones de problemas de procesamiento de información, propuestas de solución y codificación en un lenguaje de programación de alto nivel, utilizando una o más de las estructuras de datos estudiadas durante el curso.

V. DESARROLLO POR UNIDADES
UNIDAD I. Listas Enlazadas

Competencia:

Desarrollar un programa de procesamiento de información, aplicando los conceptos de listas enlazadas, con el propósito de dar solución a un problema específico pertinente, con una actitud analítica y creativa.

Contenido:

1.1 Definición

1.1.1 Listas enlazadas sencillas

1.1.2 Listas enlazadas dobles

1.1.3 Listas enlazadas circulares

1.2 Implementación y aplicaciones

1.3 Solución de problemas

Duración: 6 horas

UNIDAD II. Pilas

Competencia:

Elaborar un programa de procesamiento de información, aplicando los conceptos de pilas, con el propósito de dar solución a un problema específico pertinente, con una actitud creativa, proactiva y honesta.

Contenido:

- 2.1 Definición
- 2.2 Implementación y aplicaciones
- 2.3 Solución de problemas

Duración: 4 horas

UNIDAD III. Colas

Competencia:

Construir un programa de procesamiento de información, aplicando los conceptos de colas, con el propósito de dar solución a un problema específico pertinente, con una actitud analítica, proactiva y responsable.

Contenido:

- 3.1 Definición
- 3.2 Implementación y aplicaciones
- 3.3 Solución de problemas

Duración: 4 horas

UNIDAD IV. Árboles

Competencia:

Desarrollar un programa de procesamiento de información, aplicando los conceptos de árboles, con el propósito de dar solución a un problema específico pertinente, con una actitud analítica, creativa y honesta.

Contenido:

- 4.1 Definición
 - 4.1.1 Árboles binarios
 - 4.1.2 Árboles genéricos
 - 4.1.3 Árboles de expresiones
 - 4.1.4 Árboles de búsqueda
 - 4.1.5 Otros tipos de árboles
- 4.2 Implementación y aplicaciones
- 4.3 Solución de problemas

Duración: 6 horas

UNIDAD V. Colas con prioridad y montones.

Competencia:

Elaborar un programa de procesamiento de información, aplicando los conceptos de colas con prioridad y montones, con el propósito de dar solución a un problema específico pertinente, con una actitud creativa, proactiva y honesta.

Contenido:

- 5.1 Definiciones
 - 5.1.1 Implementación y aplicaciones de las colas con prioridad
 - 5.1.2 Montones (heaps)
- 5.2 Implementación y aplicaciones de los montones
- 5.3 Solución de problemas

Duración: 6 horas

UNIDAD VI. Tablas hash

Competencia:

Implementar un programa de procesamiento de información, aplicando los conceptos de tablas hash, con el propósito de dar solución a un problema específico pertinente, con una actitud analítica, proactiva y responsable.

Contenido:

- 6.1 Definición
- 6.2 Implementación y aplicaciones
- 6.3 Solución de problemas

Duración: 6 horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE TALLER

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
UNIDAD I				
1	Listas enlazadas	<ol style="list-style-type: none"> 1. Indaga la definición de listas enlazadas. 2. Analiza un problema real donde puede aplicarse el concepto revisado. 3. Diseña la solución del problema analizado utilizando el concepto de listas enlazadas. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Papel o cuaderno ● Lápiz o pluma 	6 horas
UNIDAD II				
2	Pilas	<ol style="list-style-type: none"> 1. Revisa la definición de pilas. 2. Analiza un problema real donde puede aplicarse el concepto visto. 3. Diseña la solución del problema revisado usando el concepto de pilas. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Papel o cuaderno ● Lápiz o pluma 	4 horas
UNIDAD III				
3	Colas	<ol style="list-style-type: none"> 1. Indaga la definición de Colas en estructura de datos. 2. Analiza un problema real donde puede aplicarse el concepto revisado. 	<ul style="list-style-type: none"> ● Papel o cuaderno ● Lápiz o pluma 	4 horas

		<ol style="list-style-type: none"> 3. Realiza el análisis de Colas FIFO y LIFO en problemas. 4. Entrega la actividad al docente para su retroalimentación. 		
UNIDAD IV				
4	Árboles	<ol style="list-style-type: none"> 1. Indaga la definición de árboles en estructura de datos. 2. Analiza un problema de la vida real donde puede aplicarse el concepto de árboles. 3. Realiza el análisis para la implementación de un programa de software que implemente árboles 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	6 horas
UNIDAD V				
5	Colas con prioridad y montones	<ol style="list-style-type: none"> 1. Revisa la definición de de colas con prioridad y montones 2. Analiza en qué situaciones de la vida real, podría implementarse esta estructura para el manejo de información. 3. Analiza y diseña un programa de software que implemente la estructura antes mencionada. 4. Entrega la actividad al docente para su 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	6 horas.

		retroalimentación. con las especificaciones que se hayan mencionado en clase.		
UNIDAD VI				
6	Tablas Hash	<ol style="list-style-type: none"> 1. Indaga la definición de tablas hash. 2. Analiza un problema de la vida real donde puede aplicarse el concepto de tablas hash. 3. Realiza el análisis para la implementación de un programa de software que implemente tablas hash. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> • Cuaderno • Lápiz o pluma 	6 horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
UNIDAD I				
1	Listas enlazadas	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación de un programa de software que implemente listas enlazadas. 2. Codifica un programa de software que sea congruente al análisis realizado. 3. Escribe el reporte de práctica correspondiente. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> • Computadora • Software para desarrollo • Compilador 	6 horas
UNIDAD II				
2	Pilas	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación de un programa de software que implemente pilas. 2. Codifica un programa de software que sea congruente al análisis realizado. 3. Escribe el reporte de práctica correspondiente. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> • Computadora • Software para desarrollo • Compilador 	4 horas
UNIDAD III				
3	Colas	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación de un programa de software para utilizar colas en estructura de datos. 	<ul style="list-style-type: none"> • Computadora • Software para desarrollo • Compilador 	4 h  oras

		<ol style="list-style-type: none"> 2. Codifica un programa de software que sea congruente al análisis realizado. 3. Escribe el reporte de práctica correspondiente. 4. Entrega la actividad al docente para su retroalimentación. 		
UNIDAD IV				
4	Árboles	<ol style="list-style-type: none"> 5. Revisa el análisis para la implementación de un programa de software que implemente árboles. 6. Codifica un programa de software que sea congruente al análisis realizado. 7. Escribe el reporte de práctica correspondiente. 8. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> • Computadora • Software para desarrollo • Compilador 	6 horas
UNIDAD V				
5	Colas con prioridad y montones	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación de un programa de software que implemente árboles. 2. Codifica un programa de software que sea congruente al análisis realizado. 3. Escribe el reporte de práctica correspondiente. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> • Computadora • Software para desarrollo • Compilador 	6 horas
UNIDAD VI				

6	Tablas hash	<ol style="list-style-type: none">1. Revisa el análisis para la implementación de un programa de software que implemente tablas hash.2. Codifica un programa de software que sea congruente al análisis realizado.3. Escribe el reporte de práctica correspondiente.4. Entrega la actividad al docente para su retroalimentación.	<ul style="list-style-type: none">• Computadora• Software para desarrollo• Compilador	6 horas
---	--------------------	--	---	---------

VII. MÉTODO DE TRABAJO

Encuadre: El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

Estrategia de enseñanza (docente):

- Técnica expositiva
- Retroalimentación
- Guía prácticas de taller y laboratorio
- Fomenta el trabajo colaborativo
- Plantea escenario de uso de las estructuras de datos
- Realiza evaluaciones
- Guía en portafolio de evidencias

Estrategia de aprendizaje (alumno):

- Codificar programas de software
- Búsqueda de información
- Búsqueda de escenario de uso de las estructuras de datos
- Realiza prácticas de taller y laboratorio
- Trabaja de manera colaborativa
- Analiza la solución a problemas utilizando estructura de datos
- Diseña programa de software
- Redacta reportes de prácticas
- Realiza portafolio de evidencias

VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

Criterios de acreditación

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir con los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

Criterios de evaluación

- Tareas y prácticas de taller.....	20%
- Exámenes parciales.....	40%
- Portafolio de evidencias.....	40%
Total.....	100%

IX. REFERENCIAS

Básicas

Downey, A. B. (2017). *Think data structures. Algorithms and information retrieval in Java*. O'Reilly.

Joyanes Aguilar, L., & Zahonero Martínez, I. (2001). *Programación en C metodología, Algoritmos y Estructura de datos*. McGraw-Hill

Joyanes, L. (2004). *Algoritmos y Estructuras de Datos. Una Perspectiva en C*. McGraw-Hill / Interamericana De España, S.A.

Karumanchi, N. (2017a). *Data structures and algorithms made easy. Data structures and algorithmic puzzles*. CareerMonk Publications.

Karumanchi, N. (2017b). *Data structures and algorithms made easy in Java. Data structures and algorithmic puzzles*. CareerMonk Publications.

Naragund, J. G., Kotre, P. A., Kanakaraddi, S., Sujata, C. (2016). *Philosophy of Data Structures in Engineering Education. International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*. <https://ieeexplore.ieee.org/document/7743168>.

Complementarias

Allen, M. (2013) *Estructura de datos en Java*. [Universidad Autónoma de Baja California]. Pearson. [clásica] <https://uabc.vitalsource.com/#/books/9788415552222/>

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd-edition). MIT Press and McGraw-Hill. [clásica]

Goodrich, M. T., Tamassia, R. y Goldwasser, M. H. (2014). *Data structures and algorithms in Java*. Wiley. [clásica]

Hubbard, J. R. (2007). *Data structures with Java*. McGraw-Hill. [clásica]

Mehta, D. P., & Sahni, S. (Eds.). (2018). *Handbook of data structures and applications*. Taylor & Francis.

X. PERFIL DEL DOCENTE

El docente que imparta la unidad de aprendizaje Estructuras de Datos debe contar con título de Ingeniero de software, Licenciado en ciencias de la computación o área afín, con conocimientos de matemáticas discretas, análisis de algoritmos y estructuras de datos; preferentemente con estudios de posgrado en ciencias de la computación y al menos dos años de experiencia docente. Debe ser proactivo, creativo y analítico.