

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:** Análisis de Algoritmos
- 5. Clave:** 40011
- 6. HC:** 02 **HT:** 01 **HL:** 02 **HPC:** 00 **HCL:** 00 **HE:** 02 **CR:** 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

Carelia Guadalupe Gaxiola Pacheco
Héctor Zatarain Aceves
Thelma Violeta Ocegueda Miramontes

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

Humberto Cervantes De Ávila
Daniela Mercedes Martínez Platas
Noemí Hernández Hernández

Fecha: 20 de febrero de 2021

II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

La finalidad de esta asignatura es comprender las bases del análisis y diseño de algoritmos para el diseño e implementación de programas de computadora eficientes en el tiempo de ejecución.

La utilidad de esta es brindar al estudiante las herramientas para la selección de algoritmos adecuados para la solución de problemas de procesamiento de información.

Se imparte en la etapa disciplinaria con carácter obligatorio. Pertenece al área de conocimiento Métodos y Tecnologías de Software

III. COMPETENCIA GENERAL DE LA UNIDAD DE APRENDIZAJE

Diseñar e implementar programas de computadora eficientes, a través de la selección de los algoritmos más adecuados, para solucionar problemas de procesamiento de información, de manera creativa, responsable y honesta.

IV. EVIDENCIA(S) DE APRENDIZAJE

Portafolio de prácticas de laboratorio que incluya reportes técnicos con descripciones de problemas de procesamiento de información, propuestas de solución y codificación en un lenguaje de programación de alto nivel, utilizando uno o más de los algoritmos analizados durante el curso.

V. DESARROLLO POR UNIDADES
UNIDAD I. Conceptos básicos de algoritmos

Competencia:

Identificar los conceptos básicos de los algoritmos, a través del análisis de sus medidas de eficiencia asintóticas, para diseñar algoritmos eficientes, con actitud crítica, analítica y sistemática.

Contenido:

- 1.1. Definición de algoritmo
- 1.2. Eficiencia de los algoritmos
 - 1.2.1. Tipos de análisis de algoritmos
 - 1.2.2. Notaciones asintóticas y orden de eficiencia
 - 1.2.3. Cálculo del orden de eficiencia de un algoritmo
- 1.3 Técnicas de diseño de algoritmos

Duración: 6 horas

UNIDAD II. Divide y vencerás

Competencia:

Analizar algoritmos donde se aplica el paradigma divide y vencerás, mediante la comparación de su complejidad e implementación, para la selección del más apropiado en la solución de un problema, con actitud reflexiva, analítica y sistemática.

Contenido:

- 2.1. El paradigma divide y vencerás
- 2.2. Algoritmos de clasificación (ordenamiento)
 - 2.2.1. Algoritmos básicos (inserción, selección y burbuja)
 - 2.2.2. Algoritmos más eficientes (mergesort, quicksort, heapsort, shellsort, radixsort)

Duración: 6 horas

UNIDAD III. Algoritmos voraces

Competencia:

Examinar la técnica voraz, mediante el análisis de la complejidad e implementación de diferentes algoritmos voraces, para la selección del más apropiado en la solución de un problema que involucre grafos, con actitud crítica, metódica y sistemática.

Contenido:

- 3.1. La técnica voraz (greedy)
- 3.2. Algoritmos voraces para grafos
 - 3.2.1. Algoritmo Kruskal
 - 3.2.2. Algoritmo Prim
 - 3.2.3. Algoritmo Dijkstra
 - 3.2.4. Heurísticas greedy

Duración: 6 horas

UNIDAD IV. Algoritmos para grafos

Competencia:

Revisar diferentes algoritmos de recorrido y búsqueda sobre grafos, mediante el análisis de su complejidad e implementación, para la selección del más apropiado en la solución de un problema de optimización, con actitud crítica, reflexiva y sistemática.

Contenido:

- 4.1. Recorrido sobre grafos
 - 4.1.1. Vuelta atrás (backtracking)
 - 4.1.2. Ramificación y poda (branch and bound)
 - 4.1.3. Árboles de juego
- 4.2 Búsqueda sobre grafos
 - 4.2.1 Búsqueda en profundidad (Depth-first search)
 - 4.2.2 Búsqueda en anchura (Breadth-first search)

Duración: 6 horas

UNIDAD V. Programación dinámica

Competencia:

Analizar el método de programación dinámica, por medio del estudio de la complejidad en problemas específicos, para comprender su utilidad y aplicación, con actitud reflexiva, metódica y crítica

Contenido:

- 5.1. Elementos de la programación dinámica
- 5.2. Problemas de programación dinámica
 - 5.2.1. Multiplicación encadenada de matrices
 - 5.2.2. Subsecuencia de mayor longitud
 - 5.2.3. Selección de actividades con pesos
 - 5.2.4. Problema de la mochila
 - 5.2.5. Caminos mínimos (Floyd y Bellman-Ford)
- 5.3. Alineación de secuencias y sus aplicaciones

Duración: 8 horas

VI. ESTRUCTURA DE LAS PRÁCTICAS DE TALLER

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
UNIDAD I				
1	Eficiencia de algoritmos	<ol style="list-style-type: none"> 1. Atiende las orientaciones del profesor para determinar la eficiencia de los algoritmos 2. Identifica las operaciones que sirven para determinar la eficiencia en diversos algoritmos de ejemplo. 3. Investiga las principales funciones que caracterizan la eficiencia de un algoritmo. 4. Grafica las principales funciones que caracterizan la eficiencia de un algoritmo. 	<ul style="list-style-type: none"> ● Computadora. ● Internet. ● Cuaderno ● Recursos bibliográficos 	2 horas
UNIDAD II				
2	Algoritmos básicos de clasificación (ordenamiento).	<ol style="list-style-type: none"> 1. Revisa algoritmos básicos de clasificación. 2. Analiza el problema provisto por el docente. 3. Diseña la solución utilizando un algoritmo básico de clasificación. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	1 hora
3	Algoritmos de clasificación (ordenamiento) más eficientes.	<ol style="list-style-type: none"> 1. Revisa algoritmos de clasificación más eficientes. 2. Analiza el problema provisto por el docente. 3. Diseña la solución utilizando un algoritmo de clasificación más eficiente. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	1 hora

		4. Entrega la actividad al docente para su retroalimentación.		
UNIDAD III				
4	Algoritmos voraces para grafos.	<ol style="list-style-type: none"> 1. Revisa algoritmos voraces para grafos. 2. Analiza el problema provisto por el docente. 3. Diseña la solución utilizando un algoritmo voraz. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	2 horas
UNIDAD IV				
5	Recorrido sobre grafos	<ol style="list-style-type: none"> 1. Revisa el método sobre recorridos de grafos. 2. Analiza el problema provisto por el docente. 3. Diseña la solución del problema usando el recorrido de grafos. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	2 horas
6	Búsqueda sobre grafos	<ol style="list-style-type: none"> 1. Revisa el método sobre búsqueda de grafos. 2. Analiza el problema provisto por el docente. 3. Diseña la solución del problema usando las técnicas de búsqueda de grafos. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	2 horas
UNIDAD V				
7	Elementos de la programación dinámica	<ol style="list-style-type: none"> 1. Revisa los elementos de la programación dinámica 2. Analiza el problema provisto por el docente. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	1 hora

		<ol style="list-style-type: none"> 3. Diseña la solución del problema usando los elementos de la programación dinámica. 4. Entrega la actividad al docente para su retroalimentación. 		
8	Problemas de programación dinámica	<ol style="list-style-type: none"> 1. Revisa los diferentes problemas de la programación dinámica como: <ol style="list-style-type: none"> a. Multiplicación encadenada de matrices b. Subsecuencia de mayor longitud c. Selección de actividades con pesos d. Problema de la mochila e. Caminos mínimos (Floyd y Bellman-Ford) 2. Analiza el problema provisto por el docente. 3. Diseña la solución del problema usando los diferentes problemas de la programación dinámica. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	4 horas
9	Alineación de secuencias y sus aplicaciones	<ol style="list-style-type: none"> 1. Revisa los elementos de la programación dinámica 2. Analiza el problema provisto por el docente. 3. Diseña la solución del problema usando los elementos de la programación dinámica. 4. Entrega la actividad al docente para su retroalimentación. 	<ul style="list-style-type: none"> ● Cuaderno ● Lápiz o pluma 	1 hora

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No.	Nombre de la Práctica	Procedimiento	Recursos de Apoyo	Duración
UNIDAD I				
1	Comparación y crecimiento de funciones asintóticas	<ol style="list-style-type: none"> 1. Atiende las orientaciones acerca del comportamiento de las funciones conforme a un conjunto de funciones planteadas. 2. Establece hipótesis sobre su crecimiento y orden de las funciones. 3. Comprueba las hipótesis utilizando un software de graficación. 4. Clasifica y ordena el conjunto de funciones de acuerdo a su orden de crecimiento (de menor a mayor). Aquellas funciones que tengan el mismo orden deben de estar en la misma categoría. 5. Escribe el reporte de práctica correspondiente. 	<ul style="list-style-type: none"> • Computadora • Software para graficación de funciones 	2 horas
UNIDAD II				
2	Algoritmos básicos de clasificación (ordenamiento).	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación de algoritmos básicos de clasificación. 2. Codifica un programa que sea congruente con el 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 • Compilador 	4 horas

3	Algoritmos de clasificación (ordenamiento) más eficientes.	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación de algoritmos de clasificación más eficientes. 2. Codifica un programa que sea congruente con el 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 • Compilador 	5 horas
UNIDAD III				
4	Algoritmos voraces para grafos.	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación de algoritmos voraces para grafos. 2. Codifica un programa que sea congruente con el 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 • Compilador 	5 horas
UNIDAD IV				
5	Recorrido sobre grafos	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación del recorrido de grafos. 2. Codifica un programa que sea congruente con el 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 • Compilador 	5 horas
6	Búsqueda sobre grafos	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación de la búsqueda de grafos. 	<ul style="list-style-type: none"> • Computadora • Compilador 	5 horas

		<ol style="list-style-type: none"> 2. Codifica un programa que sea congruente con el análisis realizado. 3. Escribe el reporte de práctica correspondiente. 4. Entrega la actividad al docente para su retroalimentación. 		
UNIDAD V				
7	Problemas de programación dinámica	<ol style="list-style-type: none"> 1. Revisa el análisis para la implementación programación dinámica. 2. Codifica un programa que sea congruente con el 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 • 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):

- Estudio de caso
- Aprendizaje basado en problemas
- Técnica expositiva
- Ejercicios prácticos
- Foros

Estrategia de aprendizaje (alumno):

- Investigación
- Estudio de caso
- Trabajo en equipo
- Exposiciones
- Organizadores gráficos
- Cuadros comparativos
- Portafolio de prácticas

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

- Evaluaciones parciales.....	30%
- Prácticas de taller	20%
- Tareas.....	10%
- Portafolio de prácticas de laboratorio con reportes técnicos.....	40%
Total.....	100%

IX. REFERENCIAS

Básicas

- Bhargava, A. Y. (2016). *Algorithms. An illustration guide for programmers and other curious people*. Estados Unidos: Manning Publications Co.
- Brassard, G. y Bratley, P. (1997). *Fundamentos de Algoritmia*. España: Prentice-Hall. [clásica].
- Kleinberg, J. y Tardos, E. (2005). *Algorithm Design*. Estados Unidos: Addison-Wesley. [clásica]
- Roughgarden, T. (2017). *Algorithms illuminated: part 1: The Basics*. Estados Unidos: Soundlikeyourself Publishing.
- Roughgarden, T. (2018). *Algorithms illuminated: part 2: Graph Algorithms and Data Structures*. Estados Unidos: Soundlikeyourself Publishing.
- Roughgarden, T. (2019). *Algorithms illuminated: part 3: Greedy Algorithms and Dynamic Programming*. Estados Unidos: Soundlikeyourself Publishing.

Complementarias

- Cormen, T. H., Leiserson, C. E., Rivest, R. L. y Stein, C. (2009). *Introduction to Algorithms*. Estados Unidos: MIT Press. [clásica].
- Fry, H. (2018). *Hello World. Being human in the age of algorithms*. Reino Unido: W. W. Norton & Company.
- Lamagna, E. A. (Junio, 2015). *Algorithmic thinking unplugged*. *Journal of Computing Sciences in College*, 30 (6), 45-52. Recuperado de <https://dl.acm.org/doi/10.5555/2753024.2753036>

X. PERFIL DEL DOCENTE

El docente que imparta la unidad de aprendizaje Análisis de Algoritmos 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.