

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
COORDINACIÓN DE FORMACIÓN BÁSICA
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA
PROGRAMA DE UNIDAD DE APRENDIZAJE

I. DATOS DE IDENTIFICACIÓN

1. Unidad Académica: Facultad de Ciencias
2. Programa (s) de estudio: Lic. en Ciencias Computacionales, Lic. en Matemáticas Aplicadas
3. Vigencia del plan:
4. Nombre de la Unidad de aprendizaje: Análisis de Algoritmos 5. Clave:
6. HC: 2 HL: 2 HT: 1 HPC: HCL: HE: 2 CR: 7
7. Etapa de formación a la que pertenece: Disciplinaria
8. Carácter de la Unidad de aprendizaje: Obligatoria X Optativa
9. Requisitos para cursar la unidad de aprendizaje:

Formuló: Dr. Everardo Gutiérrez López

Vo.Bo. Dr. Alberto Leopoldo Morán y Solares

Fecha: Septiembre de 2016

Cargo: Subdirector

II. PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

El estudio del análisis de los algoritmos representa en el perfil de los egresados de ciencias computacionales una primera etapa de abstracción y razonamiento lógico matemático para identificar las limitaciones de los modelos actuales de computación. Esto se enmarca en el área de la teoría de la computación, la cual se encarga de establecer los fundamentos de las ciencias computacionales de manera formal.

Esta asignatura tiene como finalidad que el alumno sea capaz de realizar un análisis lógico-matemático de las soluciones algorítmicas propuestas para diferentes problemáticas, así como de proponer soluciones alternativas utilizando técnicas avanzadas de diseño de algoritmos.

Esta asignatura se encuentra en la etapa disciplinaria y de carácter obligatorio para la Lic. en Ciencias Computacionales y en la etapa terminal y de carácter optativo para la Lic. en Matemáticas Aplicadas. Se recomienda el cursarla después de haber acreditado Estructura de Datos y Algoritmos.

III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE

Evaluar algoritmos utilizando las metodologías teórico-prácticas existentes para determinar su corrección y eficiencia asintótica en relación con la entrada de datos del problema, con una actitud analítica y de razonamiento lógico matemático.

IV. EVIDENCIA (S) DE DESEMPEÑO

Elabora un portafolio el cual contenga los ejercicios de análisis de las demostraciones lógico-matemáticas donde se identifique la eficiencia y corrección de los algoritmos computacionales, el portafolio debe incluir: portada, desarrollo, ejercicios, conclusión.

V. DESARROLLO POR UNIDADES

UNIDAD I. Herramientas de Análisis de Algoritmos.

Competencia: Identificar las herramientas de análisis de algoritmos para contrastar su desempeño mediante la caracterización de sus propiedades de eficiencia y corrección, de manera organizada y crítica.

Contenido

Duración: 10 horas

- I. Herramientas de Análisis de Algoritmos.
 - I.1 Definiciones e importancia de los algoritmos.
 - I.2 Corrección de un algoritmo.
 - I.3 Eficiencia en tiempo y espacio.
 - I.4 Análisis asintótico.
 - I.5 Solución de recurrencias.
 - I.6 Análisis amortizado.

UNIDAD II. Técnicas avanzadas de Diseño de Algoritmos.

Competencia: Discriminar las principales técnicas avanzadas para el diseño de algoritmos computacionales examinando sus características y aplicación a casos de estudio prácticos, con una actitud propositiva.

Contenido

Duración: 6 horas

II. Técnicas avanzadas de Diseño de Algoritmos.

- II.1 Paradigma divide y vencerás.
- II.2 Programación dinámica.
- II.3 Estrategias voraces.
- II.4 Métodos probabilísticos.
- II.5 Algoritmos de aproximación.
- II.6 Métodos Heurísticos.

UNIDAD III. Algoritmos sobre grafos.

Competencia: Distinguir los principales algoritmos para el manejo de información en grafos analizando las problemáticas que pueden representarse en esa estructura, con una actitud crítica y de abstracción.

Contenido

Duración: 6 horas

III. Algoritmos sobre grafos.

- III.1 Algoritmos elementales para grafos.
- III.2 Árboles de expansión mínima.
- III.3 Camino más corto de fuente única y múltiple.
- III.4 Redes de flujo.

UNIDAD IV. Introducción a la Teoría de la Computación.

Competencia: Detectar las características y limitaciones de los modelo computacionales realizando una abstracción de sus componentes para determinar su aplicabilidad, con una actitud crítica y de abstracción.

Contenido

Duración: 8 horas

- IV. Introducción a la Teoría de la Computación.
 - IV.1 Lenguajes Formales.
 - IV.2 Teoría de la Computabilidad.
 - IV.3 Teoría de la Complejidad Computacional.

UNIDAD V. Tópicos Selectos de Aplicación.

Competencia: Aplicar las herramientas de análisis y diseño para determinar el desempeño de los algoritmos aplicándolos a problemáticas teórico-prácticas, con una actitud crítica y propositiva.

Contenido

Duración: 2 horas

- V. Tópicos selectos de aplicación.
 - V.1 Polinomios y la FFT.
 - V.2 Teoría de números.
 - V.3 Manipulación de cadenas.
 - V.4 Geometría computacional.

VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Deducir el comportamiento de algoritmos de prueba aplicando el método de invariante de lazo para determinar su corrección, siguiendo un razonamiento lógico-matemático y una actitud crítica.	Aplicar el método de invariante de lazo en algoritmos básicos para diversas problemáticas.	Bibliografía Lápiz Papel	4 HL 2 HT
2	Identificar el desempeño de algoritmos de prueba analizando su comportamiento asintótico para determinar sus requerimientos en tiempo y espacio, con una actitud crítica y analítica.	Analizar el desempeño en tiempo y espacio de algoritmos de prueba utilizando las técnicas de análisis asintótico y de resolución de recurrencias.	Bibliografía Lápiz Papel	4 HL 2 HT
3	Aplicar algunas de las técnicas avanzadas identificando sus componentes principales para diseñar soluciones algorítmicas, de forma organizada y propositiva.	Identificar los componentes principales de las técnicas avanzadas de diseño de algoritmos para generar soluciones a problemas prácticos.	Bibliografía Lapiz Papel	6 HL 3 HT
4	Analizar estrategias algorítmicas de manipulación de información sobre grafos para determinar su desempeño mediante el estudio de problemas típicos de aplicación, de manera organizada y analítica.	Estudiar los problemas típicos de almacenamiento y manipulación de información en las estructuras de grafos.	Bibliografía Lápiz Papel	6 HL 3 HT
5	Diferenciar las características de un algoritmo computacional utilizando las herramientas de análisis para explicar su comportamiento asintótico y diferenciarlo de sus alternativas, con una actitud crítica y analítica.	Utilizar las herramientas de análisis de algoritmos exponiendo ante clase casos de ejemplo.	Bibliografía Computadora Pintarrón Plumones	4 HL 2 HT
6	Aplicar las herramientas de análisis para determinar el desempeño algoritmos evaluando soluciones computacionales a problemas teórico-prácticos, con una actitud analítica y propositiva.	Evaluar las soluciones computacionales, representadas con algoritmos, dadas a problemas teórico-prácticos.	Bibliografía Lápiz Papel	8 HL 4 HT

VII. METODOLOGÍA DE TRABAJO

Aprendizaje participativo

Durante el desarrollo de la unidad de aprendizaje, el docente fomentará la participación activa de los estudiantes en actividades individuales y grupales. Se busca facilitarle al alumno la evaluación de algunos de los principales algoritmos computacionales utilizando herramientas para su análisis y diseño, siguiendo un razonamiento lógico-matemático, manteniendo una actitud crítica, propositiva y de participación de parte de los alumnos promovido por el docente en su rol de mediador. Se sugiere incentivar la participación del alumno mediante discusiones críticas, intercambio de ideas, dinámicas de grupo, asignaciones extras que representen un reto adicional, entre otras.

Prácticas de laboratorio

En las sesiones de laboratorio, Se debe realizar la evaluación, por parte del alumno, de los principales algoritmos contenidos en la asignatura aplicados a problemas teórico-prácticos cuyo desempeño en tiempo y espacio requiera ser discriminado.

Investigación Bibliográfica

Se sugiere solicitar investigación en diferentes fuentes bibliográficas sobre temas de actualidad o temáticas que serán discutidos posteriormente en clase. El propósito de estos trabajos es fomentar el autoaprendizaje y que el estudiante aprenda a realizar investigación en medios electrónicos (Internet), libros, y revistas sobre temas del área. Las fuentes serán tanto en el idioma inglés como español. Los reportes deberán contener las referencias que se utilizaron para la realización del trabajo y debe contar imprescindiblemente una conclusión personal acerca de la investigación. El maestro debe enfatizar a los estudiantes que los reportes escritos sean claros y bien redactados, recalcándoles también las faltas de ortografía.

Ejercicios y exámenes de conocimientos

El maestro deberá aplicar al menos 2 exámenes de conocimientos durante el periodo, que permitan identificar la obtención de competencias de los estudiantes. Los exámenes podrán ser de varios tipos, tales como: de preguntas abiertas, opción múltiple y solicitud de programas. Así como, la asignación de ejercicios para ser realizados en la sesión de taller o extra clase, de tal manera que refuercen los conocimientos aprendidos durante la clase teórica. Se solicitará la entrega oportuna y formal de tareas y trabajos de investigación.

VIII. CRITERIOS DE EVALUACIÓN

Requisitos de Acreditación

Acreditar los parciales en tiempo y forma.
Entregar las tareas de programación.
Realizar las presentaciones orales.
Entregar los ejercicios de análisis lógico-matemático de cada unidad.
Cumplir con los requisitos establecidos en el estatuto escolar vigente.

Criterios de Evaluación

Los ejercicios de análisis se evaluarán considerando la utilización del razonamiento lógico-matemático para la demostración de las características de los algoritmos así como la justificación de las conclusiones obtenidas. En las presentaciones orales se valorará la capacidad de abstracción y síntesis del alumno para argumentar las propiedades de corrección y eficiencia asociadas a los algoritmos seleccionados como casos de estudio. En la evaluación de los exámenes se considerará la capacidad del alumno para utilizar las herramientas de análisis y diseño de algoritmos en la resolución de problemas teórico-prácticos.

Para la acreditación de la unidad de aprendizaje se atenderá al Estatuto Escolar Vigente, artículos 70-71, por lo que el estudiante deberá contar un mínimo de 80% de asistencias en el periodo. Tener un mínimo aprobatorio de 60 en su calificación final.

Calificación

Al menos tres exámenes parciales: 45%
Ejercicios de análisis: 30%
Tareas de programación: 10%
Presentaciones orales: 15%

IX. BIBLIOGRAFÍA

Básica

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press, 3ª Ed., 2010.
- Michael Sipser. Introduction to the Theory of Computation. Cengage Learning, 2012.
- Steven S Skiena. The Algorithm Design Manual. Springer, 2010.
- ACM Special Interest Group on Algorithms and Computation Theory.
<http://www.sigact.org/>

Complementaria

- G. Ausiello, P. Crescenzi, V. Kann, Marchetti-sp, Giorgio Gambosi, Alberto M. Spaccamela. Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties. Springer, 1999. [Clásico]
- Thomas H. Cormen. Algorithms Unlocked. The MIT Press, 2010.
- Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani. Algorithms. McGraw-Hill, 2006.
[Clásico]<http://beust.com/algorithms.pdf>
- Michael R. Garey, David S. Johnson. W. H. Freeman. Computers and Intractability: A Guide to the Theory of NP-Completeness. 1979. [Clásico]
- Michael Mitzenmacher. Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005. [Clásico]
- Rajeev Motwani and Prabhakar Raghavan. Randomized algorithms. Cambridge University Press, 1995. [Clásico]
- Robert Sedgewick, Kevin Wayne. Algorithms (4th Edition). Addison-Wesley Professional, 2011.
- Vijay V. Vazirani. Approximation algorithms. Springer, 2004. [Clásico]
- IEEE Computer Society. <http://www.computer.org/>

X. Perfil Docente

El docente deberá ser profesionalista en ciencias computacionales o áreas afines, además deberá tener conocimiento en las técnicas de análisis de algoritmos, en técnicas de diseño de algoritmos y en las bases de la teoría de la computación que permitan analizar los modelos computacionales históricos y vigentes.

