

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: Nivel: Lic. en Ciencias Computacionales, Lic. en Matemáticas Aplicadas

3. Vigencia del plan: _____

4. Nombre de la Unidad de aprendizaje: Estructura de Datos y Algoritmos 5. Clave:

6. HC: 2 HL: 2 HT: 1 HPC: ____ HCL: ____ HE: 2 CR: 7

7. Etapa de formación a la que pertenece: Básica

8. Carácter de la Unidad de aprendizaje: Obligatoria ____ ____ 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

En el ámbito de las ciencias computacionales, es de vital importancia el uso de estructuras que nos permitan almacenar, organizar y representar datos, así como la utilización de los algoritmos que nos ayuden a explotar esas estructuras. Esto permite a los estudiantes del programa de ciencias computacionales sentar las bases para el desarrollo de soluciones que optimicen los recursos computacionales disponibles. Lo cual contribuye a la formación de profesionales del área con un sentido de responsabilidad en el uso de los recursos a su disposición.

Esta asignatura tiene como finalidad que el alumno sea capaz de realizar una representación y manipulación más eficiente de los datos al proponer soluciones computacionales a problemáticas prácticas.

Esta asignatura se encuentra en la etapa básica y es de carácter obligatorio para la Lic. en Ciencias Computacionales y optativa para la Lic. en Matemáticas Aplicadas.

III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE

Desarrollar programas computacionales, discriminando las estructuras de datos y los algoritmos que faciliten representar y manipular la información necesaria para solucionar problemas de representación y manejo de datos, con una actitud analítica y propositiva.

IV. EVIDENCIA (S) DE DESEMPEÑO

Desarrolla programas computacionales que utilicen las estructuras de datos e implementen los algoritmos mediante un lenguaje de programación de alto nivel para la solución de problemas especificados por el docente. Los programas deben ser acompañados de reportes de los resultados obtenidos de la aplicación de las estructuras de datos y algoritmos en la solución de las problemáticas planteadas.

V. DESARROLLO POR UNIDADES

UNIDAD I. Introducción y estructuras de datos básicas.

Competencia: Identificar la importancia de las estructuras de datos básicas mediante la representación de datos simples y sus métodos de acceso para su aplicación en problemáticas prácticas que requieran soluciones computacionales, con una actitud de abstracción.

Contenido

Duración: 8 horas

I. Introducción y estructuras de datos básicas.

- I.1 Representación de tipos de datos y sus operaciones.
- I.2 Formas básicas de organización de los datos: arreglos, registros, conjuntos.
- I.3 Estructuras básicas de información: pilas, colas, listas ligadas.
- I.4 Recursividad y su simulación utilizando pilas.

UNIDAD II. Algoritmos, eficiencia y corrección.

Competencia: Determinar la importancia de los algoritmos para la resolución de problemáticas prácticas mediante la utilización de las técnicas de análisis de sus características, con una actitud crítica y analítica.

Contenido

Duración: 8 horas

II. Algoritmos, eficiencia y corrección.

- II.1 Algoritmos y su importancia.
- II.2 Corrección de un algoritmo.
- II.3 Concepto general de eficiencia en tiempo y en espacio.
- II.4 Notación asintótica para el crecimiento de funciones.

UNIDAD III. Ordenamiento y Búsqueda.

Competencia: Discriminar los algoritmos de ordenamiento y búsqueda identificando los pasos que realizan en sus diferentes casos de uso para determinar la conveniencia de su uso práctico, aprendiendo a estudiar su comportamiento por medio de la experimentación de forma metódica y analítica.

Contenido**Duración: 6 horas**

III. Ordenamiento y Búsqueda.

- III.1 Conceptos generales del problema de búsqueda.
- III.2 Ordenamientos básicos: por intercambio, por selección, y por inserción.
- III.3 Ordenamientos con el paradigma divide y vencerás.
- III.4 Métodos de búsqueda: secuencial, binaria e interpolada.

UNIDAD IV. Estructuras de datos no lineales.

Competencia: Organizar los datos de manera no lineal mediante el uso de estructuras de grafos y árboles para la resolución de problemáticas que requieran este tipo de organización en el manejo de la información, con una actitud de abstracción y propositiva.

Contenido**Duración: 6 horas**

IV. Estructuras de datos no lineales.

- IV.1 Introducción a las estructuras no lineales y al concepto de referencia.
- IV.2 Conceptos generales y tipos de grafos.
- IV.3 Árboles: tipos básicos, representaciones y recorridos.
- IV.4 Árboles de ordenamiento y búsqueda: binarios, rojo-negro, B.
- IV.5 Aplicaciones avanzadas de árboles: Códigos de Huffman, Sistemas de Bases de datos, Sistemas de Archivos.

UNIDAD V. Estructuras de datos avanzadas.

Competencia: Examinar algunas de las estructuras avanzadas de datos para mejorar el manejo de la información mediante su organización y la implementación de sus métodos de acceso, con una actitud crítica y de curiosidad intelectual.

Contenido

Duración: 4 horas

V. Estructuras de datos avanzadas.

V.1 Tablas Hash.

V.2 Montículos.

V.3 Árboles de Prefijo.

V.4 Conjuntos Disjuntos.

V.5 Tópicos selectos.

VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Identificar la forma en que se almacenan los tipos de datos simples elaborando programas de cómputo para la resolución de problemas típicos de manejo de datos, con una actitud propositiva.	Verificar que cada tipo de dato ocupa diferentes espacios de almacenamiento. Medir el tiempo de ejecución de las operaciones aritméticas básicas. Programar soluciones computacionales a problemas propuestos por el docente.	Bibliografía Computadora Lenguaje Compilador	2 HL 1 HT
2	Utilización de estructuras de datos para la solución de problemas prácticos mediante la representación de la información y la programación de sus métodos de acceso, de manera organizada y propositiva.	Utilizar estructuras de arreglos, registros, pilas, colas y listas en un programa básico de manipulación de datos.	Bibliografía Computadora Lenguaje Compilador	4 HL 2 HT
3	Examinar las propiedades de un algoritmo realizando ejercicios de análisis comparativo para la ilustración de los conceptos de corrección y eficiencia, con una actitud crítica y de abstracción.	Realizar el análisis de las propiedades de un conjunto de algoritmos utilizando las herramientas de invariante de lazo, cálculo del tiempo de ejecución y notación asintótica.	Bibliografía Computadora Lápiz Papel	4 HL 2 HT
4	Discriminar a los principales métodos de ordenamiento y búsqueda utilizándolos en el manejo de datos para la resolución de problemas prácticos, de forma organizada y crítica.	Utilizar diferentes métodos de ordenamiento y búsqueda en la solución de problemas que manejen grandes cantidades de datos.	Computadora Lenguaje Compilador	6 HL 4 HT
5	Estructurar los datos de manera no lineal para su aplicación en problemáticas de manejo de información mediante la	Implementar las operaciones básicas para el manejo de al menos una estructura árboles (árboles binarios de búsqueda, árboles rojo-	Computadora Lenguaje Compilador	8 HL 4 HT

6	<p>programación de los métodos de acceso de estructuras de grafo y árboles, de manera organizada y propositiva.</p> <p>Diseñar una solución computacional utilizando alguna de las estructuras de datos avanzadas para su aplicación en una problemática práctica, con una actitud propositiva y perseverante.</p>	<p>negro, árboles B, etc.).</p> <p>Utilizar las estructuras de datos avanzadas para la solución de problemas prácticos de almacenamiento de información, compresión de datos, o equivalentes.</p>	<p>Computadora Lenguaje Compilador</p>	<p>8 HL 3 HT</p>
---	--	---	--	----------------------

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 discriminación de la lógica contenida en las principales estructuras de datos, y los algoritmos, utilizada para la representación, almacenamiento y manipulación de los datos, en un ambiente crítico y participativo de parte de los alumnos promovido por el docente en su rol de mediador. En las actividades grupales, se sugiere incentivar la participación del alumno mediante discusiones críticas, intercambio de ideas, dinámicas de grupo, asignaciones extras que representen un reto adicional.

Prácticas de laboratorio

En las sesiones de laboratorio, e debe realizar la implementación, por parte del alumno, en un lenguaje de programación moderno de alto nivel las estructuras de datos y los algoritmos que conforman la unidad de aprendizaje aplicados a problemas prácticos que requieran de soluciones computacionales.

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 los ejercicios prácticos con cada examen parcial.

Entregar las tareas de programación.

Cumplir con los requisitos establecidos en el estatuto escolar vigente.

Criterios de Evaluación

Las tareas de programación se evaluarán considerando tres aspectos: adaptación de las estructuras de datos y algoritmos para la problemática planteada, resultados prácticos obtenidos con la implementación, presentación del programa computacional.

En la evaluación de los exámenes se considerará la capacidad del alumno para discriminar las estructuras de datos y los algoritmos que le permitan resolver los ejercicios teórico-prácticos que se le planteen. Se recomienda solicitar ejercicios prácticos como requisito de presentación de los exámenes parciales.

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 prácticos: 15%

Tareas de Programación: 40%.

IX. BIBLIOGRAFÍA

Básica

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press, 3ª Ed., 2010.
- Thomas H. Cormen. Algorithms Unlocked. The MIT Press, 2010.
- Narasimha Karumanchi. Data Structures and Algorithms Made Easy: Data Structure and Algorithmic Puzzles, Second Edition. CreateSpace Independent Publishing Platform, 444p. 2011.

Complementaria

- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. Data Structures and Algorithms. Tercera Edición. 1983. [Clásico]
- Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani. Algorithms. McGraw-Hill, 2006. <http://beust.com/algorithms.pdf> [Clásico]
- Donald E. Knuth. The Art of Computer Programming. Addison-Wesley Professional. 1997. [Clásico]
- Robert Lafore. Data Structures and Algorithms in Java. Sams, 2ª Ed., 2002 [Clásico]
- Robert Sedgewick, Kevin Wayne. Algorithms (4th Edition). Addison-Wesley Professional, 2011.
- ACM Special Interest Group on Algorithms and Computation Theory. <http://www.sigact.org/>
- 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 teóricos y prácticos en el manejo de estructuras de información y en el diseño e implementación de algoritmos para la administración de datos en dichas estructuras.