

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. Ciencias computacionales, Lic. Matemáticas Aplicadas, Lic. en Física
3. Vigencia del plan: _____
4. Nombre de la Unidad de Aprendizaje: Diseño 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: Básica
8. Carácter de la Unidad de Aprendizaje: Obligatoria X Optativa _____
9. Requisitos para cursar la Unidad de Aprendizaje: Ninguna

Formuló: Adrián Enciso Almanza

Vo.Bo. Alberto Leopoldo Morán y Solares

Fecha: Agosto de 2016

Cargo: Subdirector

II. PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

La asignatura de diseño de algoritmos se encuentra ubicada en la etapa básica y es de carácter obligatorio, incluye teoría, práctica y laboratorio, la cual consta de cuatro unidades: la primera unidad, aborda los conceptos y fundamentos de los algoritmos y programas; la segunda unidad, nos introduce en un ambiente de programación sencillo y fácil de comprender; la tercera unidad, se diseñan algoritmos utilizando pseudocódigo; la cuarta unidad, se implementan los algoritmos en un lenguaje de programación de alto nivel.

Esta asignatura, permite comprender la naturaleza de los algoritmos y programas, a través de las técnicas y estructuras básicas de programación estructurada, con la intención de proponer soluciones computacionales a problemas sencillos, conscientes de las fases que se deben de llevar a cabo para la creación de un programa de computadora.

III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE

Implementar algoritmos mediante el uso de la heurística y técnicas de programación estructurada, para dar solución a problemas poco complejos, que faciliten la apropiación y el uso de las estructuras algorítmicas existentes, con una actitud crítica, propositiva.

IV. EVIDENCIA (S) DE DESEMPEÑO

Elabora un reporte final que contenga los siguientes puntos: planteamiento del problema, análisis de la solución del problema, diagrama de flujo, pseudocódigo, código del programa en un lenguaje de programación de alto nivel, resultados y conclusiones.

V. DESARROLLO POR UNIDADES

Competencia:

Aplicar los conceptos y fundamentos de los algoritmos y programas, mediante el análisis de problemas de lógica y de propósito general para entender las fases que intervienen en el proceso de programación y la importancia de los algoritmos, con una actitud crítica y responsable.

Contenido

Duración 10 horas

1. Naturaleza de los algoritmos y programas.
 - 1.1. Conceptos básicos de programación.
 - 1.2. Análisis de problemas de lógica
 - 1.3. Etapas o pasos para crear un programa.

Competencia:

Identificar los elementos lógicos y de control en un algoritmo, utilizando un micro mundo o un mini lenguaje de programación sencillo, para la fácil integración de la estructura lógica, secuencia y orden de los algoritmos, con una actitud participativa y tolerante.

Contenido

Duración 15 horas

2. Mini lenguajes o micro mundos.
 - 2.1. Historia de los mini lenguajes.
 - 2.2. Características e importancia de los mini lenguajes.
 - 2.3. Conociendo un mini lenguaje (Karel, Logo, Scratch...).
 - 2.4. Secuencia, selección o condicionales, ciclos o iteraciones

Competencia:

Utilizar las estructuras de control y la modularidad mediante la representación de diagramas de flujo y pseudocódigo para elaborar algoritmos que tengan una estructura de fácil comprensión por el ser humano, con una actitud creativa y propositiva.

Contenido**Duración 25 horas**

3. Representación de los algoritmos en pseudocódigo.
 - 3.1. La trascendencia del modelo de Von Neuman.
 - 3.2. Constantes, variables y operadores matemáticos booleanos
 - 3.3. Operadores lógicos y operadores relacionales
 - 3.4. Secuencia, selección o condicionales, ciclos o iteraciones
 - 3.5. Subprogramas y/o procedimientos
 - 3.6. Estructura de algoritmos en pseudocódigo
 - 3.7. Representación gráfica de los algoritmos.
 - 3.8. Introducción a las estructuras de datos
 - 3.8.1. Manejo de arreglos unidimensionales
 - 3.8.2. Manejo de arreglos multidimensionales

Competencia:

Utilizar un lenguaje de programación de alto nivel aplicando las estructuras de control y técnicas de programación estructuradas para codificar algoritmos y obtener una solución a través de un programa de computo con una actitud crítica y responsable.

Contenido**Duración 30 horas**

4. Implementación de los algoritmos.
 - 4.1. Especificación de los lenguajes de programación.
 - 4.2. Conociendo un lenguaje de programación.
 - 4.3. Manejo de tipos de datos.
 - 4.4. Manejo de estructuras de control.
 - 4.5. Manejo de funciones propias del lenguaje.
 - 4.6. Características adicionales del lenguaje.

VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Identificar los pasos que intervienen en la solución de problemas, a través de la lluvia de ideas y experiencias previas, para determinar la lógica, el orden y la secuencia del algoritmo, con actitud participativa y entusiasmo.	Realizar una serie de ejercicios que permitan proponer la solución de problemas de lógica y documentar los pasos seguidos en su solución (sin utilizar las estructuras básicas de un algoritmo)	Pizarrón. Presentación en PowerPoint de conceptos y algoritmos de la vida diaria. Resolución de problemas a pizarrón con la participación de los alumnos.	2 Horas (taller)
2	Mostrar los elementos lógicos y de control de los algoritmos, utilizando un mini lenguaje con ambiente gráfico, para estructurar y ordenar los programas de cómputo, con actitud positiva y propositiva.	Realizar una serie de ejercicios que permitan practicar la solución de problemas sencillos en el micro mundo, incorporando las secuencia, selección o condicionales, ciclos o iteraciones y depuración de los algoritmos.	Pizarrón. Lecturas y resolución de problemas del Libro de mini lenguaje. Laboratorio de cómputo para la elaboración de prácticas [utilizando el simular del mini lenguaje]	4 Horas (taller) 8 Horas (laboratorio)

3	Expresar gráficamente la solución de algoritmos, mediante el uso de una herramienta de diseño, que permita ilustrar el flujo de la información y los componentes básicos que integran el diseño de algoritmos.	Realizar un algoritmo o un programa mediante un diagrama de flujo, en el cual se representen con un diagrama la secuencia de los datos o instrucciones que son expresados en el programa de karel	Pizarrón. Presentación en PowerPoint del Estado del Arte del tema. Seguimiento del libro de introducción a la programación [Luis Joyanes Aguilar] Herramienta de software para el diseño de diagramas de flujo	2 Horas (taller) 4 Horas (laboratorio)
4	Expresar un algoritmos en pseudocódigo o lenguaje natural, mediante el uso ejemplos matemáticos poco complejos, que permitan integrar los elementos básicos de programación, con actitud propositiva y creativa.	Realizar una serie de ejercicios que permitan practicar la solución de problemas, mediante la escritura de algoritmos, utilizando las estructuras básicas: secuencia, ramificación, bifurcación y subprogramas.	Pizarrón. Presentación en PowerPoint del Estado del Arte del tema. Seguimiento del libro de introducción a la programación [Luis Joyanes Aguilar] Mapas Conceptuales y lluvia de ideas. Laboratorio de cómputo, con software (matlab) para la implementación de los pseudocódigos.	4 Horas (taller) 4 Horas (laboratorio)
5	Elaborar una solución algorítmica, utilizando	Realizar una serie de ejercicios donde el alumno	Pizarrón. Presentación en	2 Horas

	<p>subprogramas o subrutinas de cómputo, para evidenciar el uso de la modularidad en la programación de computadoras, con actitud positiva y entusiasta.</p>	<p>practique la estructuración de sus programas mediante el uso de funciones, utilizando el envío y retorno de parámetros.</p>	<p>PowerPoint del Estado del Arte del tema. Seguimiento del libro de introducción a la programación [Luis Joyanes Aguilar] Mapas Conceptuales y lluvia de ideas.</p>	<p>(taller)</p> <p>4 Horas (laboratorio)</p>
6	<p>Elaborar una solución algorítmica, utilizando arreglos unidimensionales y bidimensionales para el almacenamiento de información en la programación de computadoras, con actitud positiva y creatividad.</p>	<p>Realizar una serie de ejercicios donde el alumno practique uso de estructuras de datos: arreglos unidimensionales y bidimensionales, y registros. Ejercicios como el ordenamiento de datos de un arreglo, búsqueda de un valor dentro de un arreglo, operaciones con matrices, entre otros</p>	<p>Pizarrón. Presentación en PowerPoint del Estado del Arte del tema. Seguimiento del libro de introducción a la programación [Luis Joyanes Aguilar] Mapas Conceptuales y lluvia de ideas. Línea de Tiempo Laboratorio de cómputo, con software (matlab) para la implementación de los pseudocódigos. Laboratorio de cómputo, con software (matlab) para la implementación de los pseudocódigos.</p>	<p>2 Horas (taller)</p> <p>4 Horas (laboratorio)</p>

7	Codificar los programas completos, utilizando una herramienta de programación de alto nivel, para la interacción con el algoritmo y la maquina, con honestidad y responsabilidad	Aprender el manejo de un lenguaje de programación (de preferencia C), para traducir los algoritmos a este lenguaje, y evaluar su correcto funcionamiento. Utilizar herramientas de programación, para desarrollar programas de cómputo, de problemas reales.	Selección y clasificación de herramientas de programación de computadoras. Selección de un lenguaje de programación (básico) para la implementación de los pseudocódigos. Presentación final en equipo utilizando herramientas de cómputo y powerpoint. Laboratorio de cómputo, con software (matlab) para la implementación de los pseudocódigos.	8 Horas (laboratorio)
---	--	--	--	-----------------------------

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. Mediante la discusión de las temáticas vistas en las clases teóricas y la asignación de ejercicios y prácticas en las sesiones de taller y laboratorio. En las actividades grupales el docente asignará un problema a cada equipo, el cual primeramente debe ser analizado en forma individual, posteriormente se discutirá y definirá una solución en equipo previa implementación de la misma. La participación del maestro en la aplicación de esta metodología es de mediador.

Prácticas de laboratorio

En las sesiones de laboratorio, el estudiante llevará a la práctica los conocimientos teóricos vistos en clase y los ejercicios realizados durante el taller. Se busca que los alumnos reflexionen y analicen los algoritmos que se proponen dentro del grupo, además se fomentará el intercambio de ideas, opiniones y experiencias entre alumnos y docente.

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

Para la acreditación de esta asignatura 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.

Criterio de calificación

Exámenes: teórico-práctico	40%
Prácticas y ejercicios	40%
Reporte final	20%
Total	100%

Criterios de Acreditación

- Se aplicarán al menos dos exámenes parciales durante el periodo.
- Las prácticas y ejercicios deberán ser entregados a la siguiente clase, se entregarán en forma impresa y/o electrónica, deberán contener una portada, algoritmo y conclusión.
- En el caso del trabajo final, la evaluación se dividirá en: reporte, exposición y el programa;

los puntos a evaluar serán:

Reporte (20%)

Planteamiento del problema.

Análisis del problema.

Diagrama de flujo.

Pseudocódigo del programa.

Código del programa en un lenguaje de programación de alto nivel.

Resultados.

Conclusiones.

IX. BIBLIOGRAFÍA

Básica	Complementaria
<ul style="list-style-type: none">■ Corona Nakamura, María Adriana, McGraw-Hill, (2011). Diseño de algoritmos y su codificación en lenguaje C, 1a edición.■ Joyanes Aguilar, Luis. Mc Graw Hill, (2012). Fundamentos generales de programación, 1a edición.■ Ramírez, Felipe, Alfaomega, (2007). Introducción a la programación : algoritmos y su implementación en VB.Net, C#, Java y C++,	<ul style="list-style-type: none">■ Blum, Edward K. Springer New York, 2011. [recurso electrónico] Computer Science: The Hardware, Software and Heart of It.■ Butterworth-Heinemann, (2011). [recurso electrónico] MATLAB a practical introduction to programming and problem solving, Attaway, Stormy.■ Duéñez Guzmán Edgar Alfredo, Hernández Sánchez Edgar Said, Ramírez. Ortegón Marte Alejandro, SYGMA, Arte en impresión, CIMAT. (2006). Los Dilemas de Karel.■ Kleinberg Jon – Tardos Éva, Pearson, (2006). Algorithm Design.■ Pérez López, César. Ibergarceta, (2011). MATLAB a través de ejemplos, 1a edición.

X. PERFIL DEL DOCENTE

El docente de esta asignatura deberá ser un profesionalista con formación en el área de computación o áreas afines; con experiencia en metodologías de análisis y técnicas de diseño de algoritmos, capaz de llevar a cabo el proceso de enseñanza-aprendizaje de forma participativa, con habilidades para transmitir sus conocimientos y propiciando en los alumnos el autoaprendizaje.