

**UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
COORDINACIÓN DE FORMACIÓN BÁSICA
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN
PROGRAMA DE UNIDAD DE APRENDIZAJE POR COMPETENCIAS**

I. DATOS DE IDENTIFICACIÓN

1. Unidad Académica: **Facultad de Ciencias**
2. Programa (s) de estudio: **Licenciatura en Matemáticas, Físico, Biología, Licenciado en Ciencias Computacionales**
3. Vigencia del plan: **2008-1**
4. Nombre de la Asignatura: **Diseño de Algoritmos**
5. Clave: **9814**
6. HC: **2** HL: **2** HT: **2** HE: **2** CR: **8**
7. Ciclo Escolar: **2009-1**
8. Etapa de formación a la que pertenece: **Básica (Tronco Común)**
9. Carácter de la Asignatura: **Obligatoria** **X** Optativa
10. Requisitos para cursar la asignatura:

Formuló: Dr. Luis Javier Villegas Vicencio, Dra. Selene Solorsa Calderon
L.C.C. Adrián Enciso Almanza

Fecha: Junio 2009

VoBo. Marcelo Rodríguez Meraz

Cargo: Subdirector

II. PROPÓSITO GENERAL DEL CURSO

Entrenar al estudiante en el diseño de algoritmos, a través de las técnicas algorítmicas básicas que le permitirán abordar el desarrollo de programas correctos y eficientes para resolver problemas sencillos, con conocimientos teóricos y prácticos, habilidades, experiencias y sentido crítico, todas ellas fundamentadas en teorías y técnicas sólidas, comprobadas y bien establecidas.

III. COMPETENCIA (S) DEL CURSO

Diseñar algoritmos sencillos, legibles y comprensibles, mediante el uso de la heurística y técnicas de programación estructurada, para el entendimiento de las estructuras algorítmicas existentes y su comportamiento, con una actitud crítica y propositiva.

IV. EVIDENCIA (S) DE DESEMPEÑO

Cuatro exámenes teóricos

Resolución de problemas en el taller.

Prácticas de laboratorio

Presentar un trabajo final donde se desarrolle el análisis de una problemática, se diseñe una solución algorítmica, y se implemente un programa utilizando todas las técnicas y herramientas de programación vistas en el curso.

V. DESARROLLO POR UNIDADES

Competencia: Analizar las fases que intervienen en el proceso de programación y la importancia de los algoritmos para la resolución de problemas utilizando la lógica con una actitud crítica y responsable.

Contenido

- 1.1.- Conceptos básicos de programación
 - 1.1.1.- Diferencia entre programar y codificar.
- 1.2.- Análisis de problemas de lógica
- 1.3.- La trascendencia del modelo de Von Neuman

Duración

1 semana

Competencia: Comprender los principios estéticos de la disciplina, analizar todo el proceso del diseño de un algoritmo bien estructurado para su comprensión, con una actitud participativa y responsable.

Contenido

- 2.1.- Especificación y estructura de un algoritmo
- 2.2.- El mundo del Robot Karel
 - 2.2.1. Instrucciones primitivas y programas simples.
 - 2.2.2. Crear instrucciones en Karel.
 - 2.2.3. Ejecución condicional de instrucciones.
 - 2.2.4. Ciclos en Karel.
 - 2.2.5. Programación avanzada del robot.
- 2.3. Compilador de Karel: Simulación y ejecución de algoritmos
- 2.4.- Uso de diagramas de flujo
 - 2.4.1. Elementos de un diagrama de flujo
 - 2.4.2. Representación de condicionales y ciclos
 - 2.4.3. Conectores
 - 2.4.4. Representación del diagrama de flujo con base en su algoritmo

Duración

5 semanas

Competencia: Diseñar algoritmos utilizando apropiadamente las estructuras de control y la modularidad, para elaborar el pseudocódigo con una actitud creativa y propositiva.

Contenido

- 3.1.- Constantes, variables y operadores matemáticos booleanos
- 3.2.- Operadores lógicos y operadores relacionales
- 3.3.- Secuencia, selección o condicionales, ciclos o iteraciones
- 3.4.- Subprogramas y/o procedimientos
- 3.5.- Estructura de algoritmos en pseudocódigo
- 3.6.- Introducción a las estructuras de datos
 - 3.6.1. Manejo de arreglos unidimensionales
 - 3.6.2. Manejo de arreglos multidimensionales
 - 3.6.3. Manejo de registros

Duración

5 semanas

Competencia: Codificar algoritmos a problemas reales para obtener su solución a través de un programa, implementado de manera ordenada utilizando un lenguaje de alto nivel.

Contenido temático	Duración
4.1. Programación avanzada mediante el uso de hojas de cálculo.	2 semanas
4.1.1. Introducción a la herramienta	
4.1.2. Manipulación de datos numéricos y alfanuméricos.	
4.1.3. Manejo de tablas (filas y columnas)	
4.1.4. Fórmulas y funciones.	
4.1.5. Manejo de gráficos	
4.1.6. Solución de problemas	
4.2. Programación avanzada mediante el uso de herramientas de cálculo numérico.	3 semanas
4.2.1. Introducción a la herramienta.	
4.2.2. Comandos básicos de programación.	
4.2.3. Vectores y matrices.	
4.2.4. Manejo de gráficos.	
4.2.5. Manejo de funciones estándar	
4.2.6. Crear subprogramas	
4.2.7. Solución de problemas	

VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Objetivo (s)	Descripción	Material de Apoyo	Duración
1	Practicar la solución de problemas de lógica, haciendo énfasis en el análisis y entendimiento del problema, así como, la descripción de los pasos a seguir en la solución del problema	Realizar una serie de ejercicios que permitan practicar la solución de problemas de lógica y documentar los pasos seguidos en su solución (sin utilizar las estructuras básicas de un algoritmo)	Ejercicios a realizar en el salón de clases	1 semana
2	Conocer el mundo de karel, habilidades, tareas y situaciones, la estructura general de un algoritmo en el lenguaje de karel, además el proceso de estructura en bloques, depuración paso a paso, así como, la capacidad de incrementar el número de instrucciones en karel.	Realizar una serie de ejercicios que permitan practicar la solución de problemas sencillos en el mundo del robot karel (solo con las instrucciones básica o primitivas), así también posteriormente incorporamos nuevo lenguaje con problemas que ayuden a incrementar el vocabulario de karel.	Pizarrón electrónico, Aula taller.	1 semana
3	Utilizar las condicionales que utiliza karel, incorporar las instrucciones IF-THEN, IF-THEN-ELSE, que permiten escribir programas más generales para karel.	Realizar una serie de ejercicios que permitan practicar el uso de las condicionales IF-THEN, IF-THEN-ELSE, condiciones que karel puede probar en su mundo, estos ejercicios nos van a permitir entender su funcionamiento, y saber cuando utilizar estas condicionales en un problema.	Pizarrón electrónico, Aula taller.	1 semana

4	Utilizar las instrucciones que repiten dentro del mundo de karel, las instrucciones ITERATE y WHILE aumentan enormemente lo conciso y la potencia del lenguaje de programación robot, una programación avanzada en el mundo de karel	Realizar una serie de ejercicios que permitan practicar el uso de las instrucciones que repiten ITERATE y WHILE, estas instrucciones permiten que karel disminuya el número de líneas de código para realizar una tarea, se construye un programa complejo utilizando refinación paso por paso y todas las instrucciones que se han aprendido.	Pizarrón electrónico, Aula taller. Se requiere de equipo de cómputo	1 semana
5	Diseño de diagrama de flujo con programas o algoritmos del lenguaje de programación del robot karel,	Realizar un algoritmo o un programa de karel y representarlo 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	Se les pide un acetato donde se muestra el programa de karel y otro acetato donde se muestra el diagrama de flujo correspondiente a ese programa.	1 semana
6	Practicar la solución de problemas de lógica, sobre situaciones factibles a automatizar, donde el alumno describa la solución utilizando las estructuras básicas de un algoritmo	Realizar una serie de ejercicios que permitan practicar la solución de problemas, mediante la escritura de algoritmos, utilizando las estructuras básicas: secuencia, ramificación y bifurcación. Aprender el manejo de un lenguaje de programación (de preferencia C), para traducir los algoritmos a este lenguaje, y evaluar su correcto funcionamiento.	Ejercicios a realizar en el salón de clases y en la sala de cómputo.	1 semanas

7	Practicar la solución de problemas mediante el uso de subprogramas (funciones), e identificar las ventajas del manejo de éstos.	Realizar una serie de ejercicios donde el alumno practique la estructuración de sus programas mediante el uso de funciones, utilizando el envío y retorno de parámetros.	Ejercicios a realizar en el salón de clases y en la sala de cómputo. Se requiere de equipo de cómputo	1 semanas
8	Practicar la solución de problemas mediante el uso de estructuras de datos, e identificar las ventajas del manejo de éstos.	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	Ejercicios a realizar en el salón de clases y en la sala de cómputo. Se requiere de equipo de cómputo	1 semana
9	Codificar los programas completos en una herramienta de programación de alto nivel.	Utilizar herramientas de programación, para desarrollar programas de cómputo, de problemas reales.	Sala de cómputo. Se requiere de equipo de cómputo	3 semanas

VII. METODOLOGIA DE TRABAJO

- Estudiar la importancia de los algoritmos en la solución de problemas.
- Analizar diversos problemas para plantear algunos de sus algoritmos.
- Expresar apropiadamente los pasos a seguir mediante un algoritmo para llegar a una solución del problema planteado.
- Diseñar el pseudocódigo del algoritmo elaborado al problema planteado.
- Diseñar pseudocódigos modulares para su fácil lectura y posterior reutilización.
- Codificar los pseudocódigos en un lenguaje de programación de alto nivel para interpretar la solución al problema planteado.

VIII. CRITERIOS DE EVALUACION

Criterio de calificación

- Exámenes: teóricos y prácticos 40%
- Tareas y/o Ejercicios 40%
- Trabajo final 20%

Criterio de acreditación

- Resolver cuatro exámenes parciales en tiempo y forma.
- Participaciones en clase.
- Cumplir con las tareas extra clase en tiempo y forma.
- Cumplir con las prácticas del taller.
- Cumplir con la presentación del trabajo final.

En el caso del trabajo final, la evaluación se dividirá en: reporte, exposición y el programa; los puntos a evaluar serán:

Reporte

Contenido

Planteamiento del problema
Algoritmo
Diagrama de flujo
Pseudocódigo del programa
Código del programa en un lenguaje de programación de alto nivel
Resultados
Conclusiones

Presentar el reporte escrito de forma ordenada y coherente.

Exposición

Contenido
Dominio del tema
Presentación
Expresarse en lenguaje apropiado y claro

Programa

Ejecución correcta del programa
Complejidad
Aplicación

IX. BIBLIOGRAFIA

Básica	Complementaria
<p>Fundamentos de Programación (Algoritmos y Estructuras de Datos) Luis Joyanes Aguilar McGraw Hill, 1999</p> <p>Metodología de la Programación (Algoritmos, Diagramas de Flujo y Programas) Osvaldo Cairó Alfaomega,</p> <p>Introducción Gradual a la Programación (El robot Karel) Richard E. Pattis Limusa, Noriega.</p>	<p>Computación y Programación Moderna (Perspectiva integral de la informática) Guillermo Levine Addison Wesley</p> <p>Programming in MATLAB Herniter, Marc E. Brooks/Cole-Thomson Learning,</p> <p>Excel 2003 formulas Walkenbach, John. Wiley</p>