

# UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

COORDINACIÓN DE FORMACIÓN BÁSICA

COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA

## PROGRAMA DE UNIDAD DE APRENDIZAJE HOMOLOGADO

### I. DATOS DE IDENTIFICACIÓN

1. Unidad académica (s): **CAMPUS ENSENADA:** FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO, FACULTAD DE INGENIERÍA Y NEGOCIOS SAN QUINTÍN  
**CAMPUS MEXICALI:** FACULTAD DE INGENIERÍA, ESCUELA DE INGENIERÍA Y NEGOCIOS GUADALUPE VICTORIA  
**CAMPUS TIJUANA:** FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA, FACULTAD DE INGENIERÍA Y NEGOCIOS
2. Programa (s) de estudio: (Técnico, Licenciatura (s)): INGENIERÍA EN COMPUTACIÓN
3. Vigencia del plan: 2009-2
4. Nombre de la unidad de aprendizaje Algoritmos y Estructura de Datos
5. Clave: 12098
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 : Ninguna

## Firmas Homologadas

Fecha de elaboración: Revisión mayo de 2013

Formuló:

Alma Leticia Palacios Guerrero  
Cecilia Margarita Curlango Rosas  
María Luisa González Ramírez  
Gloria Etelbina Chávez Valenzuela  
Pedro Núñez Yepiz

Vo. Bo.

Dr. Daniel Hernández Balbuena

Cargo: Subdirector Facultad de Ingeniería (Mexicali)

Vo. Bo

Q. Noemí Hernández Hernández

Cargo: Subdirectora Facultad de Ciencias Químicas e Ingeniería (Tijuana)

Vo. Bo

M.I. Joel Melchor Ojeda Ruiz

Cargo: Subdirector Facultad de Ingeniería, Arquitectura y Diseño (Ensenada)

Vo. Bo

M.C. Lizzette Velasco Aulcy

Cargo: Subdirectora Facultad de Ingeniería y Negocios San Quintín (San Quintín)

Vo. Bo

Dra. Ana María Vázquez Espinoza

Cargo: Subdirectora Escuela de Ingeniería y Negocios Guadalupe Victoria (Guadalupe Victoria)

UNIVERSIDAD AUTONOMA  
DE BAJA CALIFORNIA



FACULTAD DE INGENIERIA,  
ARQUITECTURA Y DISEÑO  
ENSENADA, B.C.

UNIVERSIDAD AUTONOMA  
DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS  
QUÍMICAS E INGENIERÍA

UNIVERSIDAD AUTONOMA  
DE BAJA CALIFORNIA



FACULTAD DE  
INGENIERIA

## **II. PROPÓSITO GENERAL DEL CURSO**

Que el alumno utilice comprenda la aplicación y comportamiento de las estructuras de datos. Además conocerá las ventajas y desventajas de algunos algoritmos de ordenamiento y búsqueda.

## **III. COMPETENCIA (S) DEL CURSO**

Eficientar los sistemas de procesamiento de información seleccionando algoritmos y estructuras de datos acordes a las necesidades de los sistemas

## **IV. EVIDENCIA (S) DE DESEMPEÑO**

Realizar una aplicación de cómputo basado en el uso de algoritmos eficientes con la mejor estructura de datos que resuelva el problema.

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

El alumno será capaz de identificar las diferentes estructuras de datos y aplicar una de ellas en la solución de problemas con agilidad y congruencia

### CONTENIDO HORAS

**DURACIÓN 2**

Unidad 1 Introducción a las estructuras de datos

1.1 Concepto de estructura de datos

1.2 Clasificación de las estructuras de datos

1.3 Implementación de estructuras de datos en un lenguaje orientado a objetos.

1.3.1 Representación de una estructura de datos por medio de clases

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

El alumno será capaz de identificar y aplicar las diferentes estructuras de datos lineales en la solución de problemas con agilidad y congruencia

.

### CONTENIDO HORAS

**DURACIÓN 14**

#### Unidad 2 Pilas, Colas y Listas

##### 2.1 Pilas

2.1.1 Definición

2.1.2 Representación

2.1.3 Manejo

2.1.4 Recursión

##### 2.2 Colas

2.2.1 Definición

2.2.2 Representación

2.2.3 Manejo

##### 2.3 Listas

2.3.1 Definición

2.3.2 Representación

2.3.3 Manejo

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

El alumno entenderá los métodos de ordenación y búsqueda visto en clase para aplicarlos correctamente en la solución de problemas de computación.

### CONTENIDO HORAS

**DURACIÓN 10**

#### Unidad 3 Ordenamientos y Búsquedas

##### 3.1 Complejidad de los algoritmos.

###### 3.1.1 Complejidad de espacio

###### 3.1.2 Complejidad de tiempo.

##### 3.2 Definición de ordenamiento interno y externo

##### 3.3 Métodos de ordenamiento

###### 3.3.1 Inserción y selección directa

###### 3.3.2 Shell

###### 3.3.3. Quicksort

###### 3.3.4 Ordenamiento externo

###### 3.3.4.1 Mezcla

###### 3.3.4.2 Mezcla equilibrada

##### 3.4 Búsquedas

###### 3.4.1 Búsqueda binaria

###### 3.4.2 Búsqueda hash o por transformación de claves

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

Programar aplicaciones de cómputo que utilicen múltiples hilos para la ejecución de tareas de manera simultánea respetando la integridad de los datos utilizados.

### CONTENIDO HORAS

**DURACIÓN 6**

#### Unidad 4 Arborecencias y Grafos

##### 4.1 Arboles binarios

4.1.1 Definición de árbol binario y sus componentes.

4.1.2 Definición de árbol binario completo, árbol binario lleno y montículo

4.1.3 Árbol binario de búsqueda

4.1.4 Recorridos de árboles.

4.1.4.1 Inorden

4.1.4.2 Postorden

4.1.4.3 Preorden

4.1.5 Representación de expresiones aritméticas y tokens.

##### 4.2 Definición de árboles B

##### 4.3 Grafos

4.3.1 Definición de términos relacionados con grafos

4.3.1.1 Grafos dirigidos

4.3.1.2 Grafos no dirigidos

4.3.1.3 Vértices

4.3.1.4 Aristas

4.3.1.5 Camino

4.3.1.6 Grafo completo

4.3.1.7 Grafo con peso

4.3.1.8 Matriz de adyacencia

4.3.1.9 Listas de adyacencia

Cuarto Parcial en periodo de ordinarios



## VI. ESTRUCTURA DE LAS PRÁCTICAS

No. de Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Reafirmar los conocimientos de la POO diseñando y construyendo programas que utilicen clases	Plantear un problema que requiera para su resolución clases y arreglos dinámicos de objetos	Equipo de cómputo con JDK instalado y práctica #1 impresa.	2 horas
2	el alumno construirá programas que utilicen pilas para resolver problemas	El alumno creará un programa que utilice pilas para resolver un problema de aplicación. En dicho programa el alumno deberá utilizar la clase Stack en donde se pueden almacenar diferentes tipos de datos	Equipo de cómputo con JDK y práctica #2 impresa.	2 horas
3	Utilizar la recursividad en programas para resolver problemas y analizar los efectos de la recursividad en los programas para poder determinar los casos en que es apropiado el uso de ella	El alumno implementará dos soluciones a un problema planteado. Una de éstas será utilizando algoritmos recursivos y la otra no utilizará recursividad	Equipo de cómputo con JDK y práctica #2 impresa.	2 horas
4	El alumno estructurará programas en los que se utilicen colas para almacenar datos	El alumno creará un programa que plantee una solución a un problema real en el que utilice colas para almacenar los datos	Computadora con el JDK instalado y acceso a la documentación del API de Java.	2 horas
5	El alumno creará programas que resuelvan problemas utilizando como estructura de datos principal	El alumno utilizará la clase List dentro de un programa de aplicación	Computadora con el JDK instalado y acceso a la	2 horas

	las listas ligadas lineales		documentación del API de Java.	
6	El alumno creará programas que resuelvan problemas utilizando como estructura de datos principal las listas doblemente ligadas lineales	El alumno implementará una clase que represente una lista doblemente ligada lineal y la utilizará dentro de un programa de aplicación	Computadora con el JDK instalado y acceso a la documentación del API de Java.	2 horas
7	El alumno creará programas que resuelvan problemas utilizando como estructura de datos principal las listas ligadas circulares	El alumno implementará una clase que represente una lista doblemente ligada circular y la utilizará dentro de un programa de aplicación	Computadora con el JDK instalado y acceso a la documentación del API de Java.	4 horas
8	El alumno creará programas que implementen distintos algoritmos de ordenamiento para comprobar su eficiencia	El alumno producirá programas que implementen algunos de los algoritmos de ordenamiento como inserción y selección directa, shell y quicksort y medirán su eficiencia para constatar los datos presentados en clase sobre estos	Computadora con el JDK instalado y acceso a la documentación del API de Java	4 horas
9	El alumno creará programas que implementen distintos algoritmos de búsquedas para comprobar su eficiencia	El alumno producirá programas que implementen los algoritmos de búsqueda como búsqueda binaria o búsqueda hash	Computadora con el JDK instalado y acceso a la documentación del API de Java	4 horas
10	el alumno diseñará e implementará un programa de aplicación en el que se utilicen las estructuras de	el alumno realizará un programa de mediana complejidad en el que se utilicen las buenas practicas de	Computadora con el JDK instalado y acceso a la	4 horas

	datos de los algoritmos estudiados en la clase para integrar todos los conocimientos adquiridos durante el curso	la programación y se utilicen correctamente las estructuras de datos y los algoritmos de búsqueda y ordenación que se estudiaron en clase	documentación del API de Java	
--	--	---	-------------------------------	--

## **VII. METODOLOGÍA DE TRABAJO**

El alumno realizará lecturas previas.

Los temas serán presentados por el docente en clase, ejemplificando los conceptos mediante pseudocódigo y fragmentos de código elaborado en lenguaje Java.

El alumno resolverá ejercicios prácticos aplicando los conceptos vistos en clase y se utilizarán técnicas grupales buscando siempre la participación activa de los alumnos.

### VIII. CRITERIOS DE EVALUACIÓN

CALIFICACIÓN	ACREDITACIÓN	EVALUACIÓN
<p>La calificación final se compondrá de la siguiente manera:</p> <p>Primer parcial ( teórico)      25%</p> <p>Segundo parcial ( teórico)    25%</p> <p>Tercer parcial ( teórico)      25%</p> <p>Cuarto parcial (teórico)      25%</p> <p>-----</p> <p>70%</p> <p>Prácticas de laboratorio</p> <p>30%</p> <p>-----</p> <p><b>100%</b></p>	<p>Tener un porcentaje mínimo de 80% de asistencia.</p> <p>Tener un promedio mayor o igual a 60.</p> <p>Tener acreditado el laboratorio.</p>	<p>Al término de cada unidad se realizarán dinámicas para verificar la correcta aplicación del conocimiento.</p> <p>En el caso de los exámenes prácticos se fomentará el trabajo en quipo y se realizará una retroalimentación después de la revisión de los mismos.</p>

## IX. BIBLIOGRAFÍA

Básica	Complementaria
<p>Core Java, Vol. 1 y 2, Eighth edition. Cay S. Horstmann, Gary Cornell. Prentice-Hall. 2007.</p> <p>Java Swing, second edition. Marc Loy, Robert Eckstein, Dave Wood, James Elliott, Brian Cole. O'Reilly. 2002.</p> <p>Java Network Programming, Third Edition. Elliotte Harold, Elliotte Rusty Harold. O'Reilly. 2004.</p> <p>Java I/O. Elliotte Harold. O'Reilly. 2006.</p>	<p>Java How to program Fourth Edition Deitel &amp; Deitel Prentice Hall, 2002</p> <p>Java con Programación Orientada a Objetos y aplicaciones en la WWW Paul S. Wang Editorial Thomson, 2000</p>