



Casa abierta al tiempo

UNIVERSIDAD AUTONOMA METROPOLITANA

PROGRAMA DE ESTUDIOS

UNIDAD	IZTAPALAPA	DIVISION	CIENCIAS BASICAS E INGENIERIA	1 / 3
NOMBRE DEL PLAN LICENCIATURA EN COMPUTACION				
CLAVE	UNIDAD DE ENSEÑANZA-APRENDIZAJE		CRED.	11
2151105	ALGORITMOS Y PATRONES DE ALMACENAMIENTO NO LINEALES ORIENTADOS A OBJETOS		TIPO	OBL.
H.TEOR. 4.0	SERIACION 2151104		TRIM. IV	
H.PRAC. 3.0				

OBJETIVO(S):

Objetivo General:

Que al final de la UEA el alumno sea capaz de:

- Resolver problemas mediante programas desarrollados haciendo uso de un proceso definido, que usen patrones de almacenamiento no lineales con el paradigma de orientación a objetos.

Objetivos Específicos:

Que al final de la UEA el alumno sea capaz de:

- Aplicar la programación orientada a objetos (nivel avanzada).
- Aplicar la programación orientada a eventos.
- Aplicar patrones no lineales de almacenamiento de datos.
- Conocer y comprender la implementación de patrones de almacenamiento no lineales.

CONTENIDO SINTETICO:

1. Programación orientada a objetos avanzada (1.5 semanas).
 - 1.1 Herencia.
 - 1.2 Polimorfismo.
 - 1.3 Clases abstractas.
 - 1.4 Clases concretas.
2. Programación orientada a eventos (1.5 semanas).
 - 2.1 Control y manejo de eventos.
 - 2.2 Interfaces del tipo ventana.



UNIVERSIDAD AUTONOMA METROPOLITANA

APROBADO POR EL COLEGIO ACADEMICO
EN SU SESION NUM. 354

EL SECRETARIO DEL COLEGIO

CLAVE 2151105

ALGORITMOS Y PATRONES DE ALMACENAMIENTO NO LINEALES
ORIENTADOS A OBJETOS

- 2.3 Relación de eventos con el código de su procesamiento.
3. Estrategias personales de calidad (1 semana).
- 3.1 Revisión como estrategia de calidad.
- 3.2 Defectos producidos por condiciones límite.
4. Patrones no lineales de almacenamiento de datos (3 semanas).
- 4.1 Árboles binarios de búsqueda.
- 4.2 Árboles n-arios.
- 4.3 Gráficas.
5. Patrones (2 semanas).
- 5.1 Patrones de diseño.
- 5.2 Patrones de programación.
6. Implementación de patrones de almacenamiento (2 semanas).
- 6.1 Implementación de listas.
- 6.2 Implementación de árboles.

MODALIDADES DE CONDUCCION DEL PROCESO DE ENSEÑANZA-APRENDIZAJE:

El profesor utilizará la clase magistral para exponer los temas del curso propiciando la participación activa y corresponsable en el proceso de enseñanza-aprendizaje, el pensamiento crítico, la disciplina y el rigor en el trabajo académico, así como la capacidad para aprender por sí mismo.

Para lograr lo anterior se podrán desarrollar actividades tales como tareas de programación, trabajos de investigación y exposición de temas. Se deberán desarrollar exhaustivamente ejemplos y ejercicios sobre los temas abordados. Para las actividades de programación se deberá elegir un lenguaje de programación orientada a objetos, por ejemplo Java, C#, Python, C++, etc.

En las horas de práctica, se utilizará la modalidad de sesiones de laboratorio de cómputo para que los alumnos resuelvan problemas de manera individual o grupal.

MODALIDADES DE EVALUACION:

Evaluación Global:

La evaluación global de esta UEA incluirá evaluaciones periódicas y, a juicio del profesor, una evaluación terminal.



UNIVERSIDAD AUTONOMA METROPOLITANA

APROBADO POR EL COLEGIO ACADEMICO
EN SU SESION NUM. 354

[Handwritten Signature]
EL SECRETARIO DEL COLEGIO

CLAVE 2151105

ALGORITMOS Y PATRONES DE ALMACENAMIENTO NO LINEALES
ORIENTADOS A OBJETOS

Se sugiere que las evaluaciones periódicas sean un mínimo de dos escritas y una oral.

El profesor seleccionará los elementos de evaluación periódica de entre las siguientes: evaluaciones, participación en clase, tareas de programación, trabajos de investigación, presentaciones de temas y programas realizados en el laboratorio.

Los factores de ponderación quedarán a juicio del profesor y se darán a conocer al inicio del curso.

Evaluación Recuperación:

A juicio del profesor, consistirá en una evaluación escrita que incluya todos los contenidos teóricos y prácticos de la UEA, o sólo aquellos que no fueron cumplidos durante el trimestre.

BIBLIOGRAFIA NECESARIA O RECOMENDABLE:

1. Alen, M. (2000). Estructuras de Datos en Java, Prentice-Hall, EUA.
2. Dale, N., Joyce, D., Weems, C., (2002), Object-Oriented Data Structures Using Java, Jones & Bartlett Learning, EUA.
3. Faison, T. (2006), Event-Based Programming: Taking Events to the Limit, Apress; 1a. Ed., EUA.
4. Guardati Bueno, S., (2007), Estructuras de Datos Orientadas a Objetos: Algoritmos en C++, Pearson, México.
5. Jones, Bartlett, Wiener, R., Pinson, L., (2000), Fundamentals of OOP and Data Structures in Java; Cambridge University Press, Inglaterra.
6. Penton, R., (2003), Data Structures for Game Programmers; Premier Press, EUA.



UNIVERSIDAD AUTONOMA METROPOLITANA

APROBADO POR EL COLEGIO ACADEMICO
EN SU SESION NUM. 354
EL SECRETARIO DEL COLEGIO