

## ESTRUCTURAS DE DATOS

<b>MATERIA:</b>	Estructuras de Datos
<b>CLAVE Y GRUPO:</b>	TC1018.01
<b>HORARIO:</b>	Lu-Ju 13:00-14:25
<b>SALÓN:</b>	5303
<b>MODALIDAD DEL CURSO:</b>	Curso rediseñado Tec Blackboard utilizando como técnica didáctica el Aprendizaje basado en problemas
<b>PROFESOR:</b>	Dr. Vicente Cubells
<b>CORREO ELECTRÓNICO:</b>	vcubells@itesm.mx
<b>UBICACIÓN:</b>	Sala de profesores Aulas 1 Nivel 2
<b>HORARIO DE ASESORÍA:</b>	Previa cita, horario abierto

### OBJETIVO GENERAL

Al finalizar este curso, el alumno será capaz de dar solución a problemas planteados a través de la construcción de software que hace uso de algoritmos y estructuras de datos de manera eficiente.

### OBJETIVOS ESPECÍFICOS

#### 1. Abstracción de datos

- Aprender a utilizar la abstracción de datos, la abstracción procedural y a ocultar la información para manejar la complejidad.
- Comprender cómo los tipos de datos abstractos se emplean durante la construcción de un sistema computacional.
- Aprender a implementar un tipo de datos abstracto haciendo uso de clases.
- Entender los diferentes niveles de abstracción y cómo se emplean en el desarrollo de software.

#### 2. Recursión

- Conocer el concepto de recursividad.
- Saber cuando aplicar recursividad en un problema.
- Entender las ventajas y desventajas de una solución recursiva.

#### 3. Algoritmos de búsqueda

- Conocer e implementar el algoritmo de búsqueda secuencial en arreglos.
- Conocer e implementar el algoritmo de búsqueda binaria en arreglos.

#### 4. Algoritmos de ordenamiento

- Entender la importancia de ordenar la información.
- Conocer los principales algoritmos de ordenamiento.
- Implementar los algoritmos de burbuja, selección e inserción directa.
- Comprender el funcionamiento del algoritmo Merge Sort y cómo puede ser implementado.
- Comprender el funcionamiento del algoritmo Quick Sort.

#### 5. Manejo de memoria

- Conocer el uso de apuntadores y/o referencias en el contexto de un programa.
- Conocer a grandes rasgos cómo se administra la memoria de un programa dentro de su ambiente de ejecución.

#### **6. Estructuras de datos lineales**

- Conocer las características de la estructura lista encadenada en todas sus variantes.
- Implementar la lista encadenada lineal y la lista encadenada ordenada.
- Aprender a construir la lista doblemente encadenada.
- Entender las características y usos de las estructuras pilas y filas.
- Implementar las estructuras pilas y filas en arreglos, listas lineales y listas circulares.
- Conocer el funcionamiento de la estructura cola.

#### **7. Estructuras de datos no lineales**

- Conocer la estructura árbol binario y sus aplicaciones.
- Entender las ventajas que ofrece un árbol binario con respecto a las listas lineales.
- Implementar un árbol binario de búsqueda.
- Conocer e implementar las diferentes maneras en las que se puede recorrer un árbol binario.
- Conocer la estructura grafo y sus aplicaciones.
- Comprender las diferentes maneras en las que se puede implementar un grafo.
- Conocer e implementar los siguientes algoritmos de recorridos de grafos: primero en profundidad y primero en amplitud.
- Conocer la técnica de hashing y sus ventajas en la búsqueda de información.
- Utilizar la técnica de hashing para implementar conjuntos y mapas (diccionarios o arreglos asociativos).

### **COMPETENCIAS QUE DESARROLLA EL CURSO:**

- Habilidad para diseñar e implementar soluciones de software, para problemas manejables en computadora, utilizando abstracción y estructuras de datos.

### **CONTENIDO TEMÁTICO**

#### **Tema 1. Abstracción de datos**

Abstracción de datos  
Tipos de datos abstractos  
Niveles de abstracción

#### **Tema 2. Recursión**

Definición de recursión  
Uso de recursión en la solución a un problema

#### **Tema 3. Algoritmos de búsqueda**

Búsqueda secuencial  
Búsqueda binaria

#### **Tema 4. Algoritmos de ordenamiento**

Algoritmos de ordenamiento simples  
Merge Sort  
Quick Sort

#### **Tema 5. Manejo de memoria**

Administración de memoria  
Apuntadores y referencias

## **Tema 6. Estructuras de datos lineales**

Listas encadenadas (lineales, circulares, dobles y ordenadas)  
Pilas  
Filas  
Colas

## **Tema 7. Estructuras de datos no lineales**

Árboles binarios  
Grafos  
Técnica de hashing

## **RECURSOS DIDÁCTICOS**

### • **LIBROS DE TEXTO**

1. Paul Deitel and Harvey Deitel (2014). C++ How to Program. Prentice Hall. 9 Edition. ISBN 978-0133378719.
2. Paul Deitel and Harvey Deitel (2014). C++11 for Programmers. Prentice Hall. 2 Edition. ISBN 978-0133439854.
3. Adam Drozdek (2012). Data Structures and Algorithms in C++. Cengage Learning; 4 Edition. ISBN 978-1133608424.
4. Narasimha Karumanchi (2011). Data Structures and Algorithms Made Easy: Data Structure and Algorithmic Puzzles. CreateSpace Independent Publishing Platform; 2 Edition. ISBN 978-1468108866.
5. Robert Sedgewick and Kevin Wayne (2011). Algorithms. Addison-Wesley Professional; 4th edition. ISBN 978-0321573513.
6. Koffman, Elliot B., Objects, abstraction, data structures, and design : using C++ / Elliot B. Koffman y Paul A.T. Wolfgang., , Hoboken, NJ : John Wiley & Sons, Inc., c2006., , , [9780471467557 (rústica)], [0471467553 (encuadernado)]
7. Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language (2nd Edition), Prentice Hall PTR, ISBN 978-0131103627

### • **MATERIALES DE APOYO**

1. Paul Deitel. C++ Fundamentals Part I and II. Disponible en: <https://www.udemy.com/c-2008-fundamentals/?couponCode=Deitel30&siteID=YSqondvmOXo-bH4qeKVaijUoheUXgAv70w>
2. The C++ Resources Network. Disponible en <http://www.cplusplus.com>
3. Silicon Graphics International(2014). Standard Template Library Programmer's Guide. Disponible en: <https://www.sgi.com/tech/stl/>
4. Algorithms and data structures in C/C++. Disponible en: <http://www.cprogramming.com/algorithms-and-data-structures.html>

### • **IDE**

1. Xcode. <https://developer.apple.com/xcode/>
2. CodeLite. <http://codelite.org>
3. Code::Blocks. <http://www.codeblocks.org/>
4. Eclipse for C/C++. <http://www.eclipse.org/>

5. Visual Studio Express for Windows Desktop. <http://www.microsoft.com/express/>
6. K Develop IDE. <https://www.kdevelop.org>

- **PLATAFORMA TECNOLÓGICA**

El material del curso, así como las actividades a desarrollar están disponibles en la plataforma Blackboard.

## CALENDARIZACIÓN DE ACTIVIDADES DEL SEMESTRE

Sem.	Fecha	Temas, subtemas y objetivos específicos	Tareas y/o actividades extraclase
1	12/01/15 15/01/15	<b>Tema 1.</b> Abstracción de datos Abstracción de datos Tipos de datos abstractos Niveles de abstracción <b>Proyecto Final. Orientación</b> <b>Tema 2.</b> Recursión Definición de recursión Uso de recursión en la solución a un problema	Estudio independiente <b>Trabajar en el Proyecto final</b>
2	19/01/15 22/01/15	<b>Tema 2.</b> Recursión Definición de recursión Uso de recursión en la solución a un problema <b>Tarea 1. Orientación</b>	Estudio independiente <b>Trabajar en el Proyecto final</b>
3	26/01/15 29/01/15	<b>Tema 3.</b> Algoritmos de búsqueda Búsqueda secuencial Búsqueda binaria	Estudio independiente <b>Entregar Tarea 1</b> <b>Trabajar en el Proyecto final</b>
4	02/02/15 05/02/15	<b>Asueto</b> <b>Tema 4.</b> Algoritmos de ordenamiento Algoritmos de ordenamiento simples Inserción Selección <b>Tarea 2. Orientación</b>	Estudio independiente <b>Trabajar en el Proyecto final</b>
5	09/02/15 12/02/15	<b>Tema 4.</b> Algoritmos de ordenamiento Quick Sort Merge Sort Ejercicios	Estudio independiente <b>Entregar Tarea 2</b> <b>Trabajar en el Proyecto final</b>
6	16/02/15 19/02/15	<b>Tema 5.</b> Manejo de memoria Administración de memoria Apuntadores y referencias <b>Primer parcial</b>	Estudio independiente <b>Trabajar en el Proyecto final</b>
7	23/02/15 26/02/15	<b>Retroalimentación del primer parcial</b> <b>Tema 5.</b> Manejo de memoria Administración de memoria Apuntadores y referencias <b>Tarea 3. Orientación</b>	Estudio independiente <b>Trabajar en el Proyecto final</b>
8	02/03/15 05/03/15	<b>Tema 6.</b> Estructuras de datos lineales Listas encadenadas (lineales, circulares, dobles y ordenadas) <b>SEMANA i</b> (No hay clases pero deben participar en las actividades ya que representarán el 7% de la calificación de la materia)	Estudio independiente <b>Entregar Tarea 3</b> <b>Trabajar en el Proyecto final</b>
9	09/03/15 12/03/15	<b>Tema 6.</b> Estructuras de datos lineales Listas encadenadas (lineales, circulares, dobles y ordenadas) <b>Tarea 4. Orientación</b>	Estudio independiente <b>Trabajar en el Proyecto final</b>

Sem.	Fecha	Temas, subtemas y objetivos específicos	Tareas y/o actividades extraclase
10	16/03/15 19/03/15	<b>Asueto</b> <b>Tema 6.</b> Estructuras de datos lineales Pilas	Estudio independiente <b>Entregar Tarea 4</b> <b>Trabajar en el Proyecto final</b>
11	23/03/15 26/03/15	<b>Tema 6.</b> Estructuras de datos lineales Colas <b>Tarea 5. Orientación</b>	Estudio independiente <b>Trabajar en el Proyecto final</b>
12	30/03/15 02/04/15	Vacaciones por Semana Santa	Estudio independiente <b>Trabajar en el Proyecto final</b>
13	06/04/15 09/04/15	<b>Tema 7.</b> Estructuras de datos no lineales Árboles binarios <b>Segundo parcial</b>	Estudio independiente <b>Entregar Tarea 5</b> <b>Trabajar en el Proyecto final</b>
14	13/04/15 16/04/15	<b>Retroalimentación del segundo parcial</b> <b>Tema 7.</b> Estructuras de datos no lineales Árboles binarios <b>Tarea 6. Orientación</b>	Estudio independiente <b>Trabajar en el Proyecto final</b>
15	20/04/15 23/04/15	<b>Tema 7.</b> Estructuras de datos no lineales Grafos	Estudio independiente <b>Entregar Tarea 6</b> <b>Trabajar en el Proyecto final</b>
16	27/04/15 30/04/15	<b>Tema 7.</b> Estructuras de datos no lineales Grafos	Estudio independiente <b>Trabajar en el Proyecto final</b>
17	04/05/15	<b>Proyecto final. Presentación</b>	Estudio independiente

## ENTREGA DE TAREAS

- Cada tarea tendrá asignada una fecha y hora de entrega para poder ser calificada sobre 100.
- No se aceptan tareas fuera del día y hora señaladas.
- Las tareas se suben a Blackboard y no se aceptan por e-mail.
- Las tareas relacionadas con problemas de programación se calificarán de la siguiente manera:
  - Deberá subirse a GitHub el código fuente de los programas, si no se sube el código fuente la calificación será cero (0) puntos.
  - Si el programa no compila, la calificación será cero (0) puntos.
  - Si el programa compila, la calificación estará basada en el porcentaje de salidas correctas del mismo.

## PROYECTO FINAL

La estructura del curso está basada en Aprendizaje Basado en Problemas (PBL, por sus siglas en inglés), los conceptos vistos en el salón de clases serán puestos en práctica por los estudiantes en el proyecto final que realizarán a lo largo del semestre.

Como se desarrollará el proyecto con software libre, el código del mismo deberá subirse a un sitio de control de versiones como GitHub para que quede disponible a toda la comunidad y a toda aquella persona que quiera hacer uso del mismo.

Requisitos:

- Se desarrollará en equipos de dos (2) personas.
- La codificación se realizará en C++.
- La complejidad del proyecto formará parte de su calificación.
- La entrega final incluye una presentación del proyecto en la última semana de clases.
- La calificación del proyecto final se obtiene de la siguiente manera:

Concepto	%
Originalidad de la solución	5%
Complejidad del software	10%
Funcionalidad y operatividad del software	50%
Calidad del software	20%
Presentación y exposición (incluye su subida a un sitio de código open source)	15%
<b>Proyecto final</b>	<b>100%</b>

## SISTEMA DE EVALUACIÓN DEL SEMESTRE

Fechas de los exámenes parciales y final:

Examen	Fecha
Primer parcial	Jueves 19 de Febrero de 2015
Segundo parcial	Jueves 09 de Abril de 2015
<b>Examen final</b>	<b>Martes 12 de Mayo de 2015 a las 12:00 horas</b>

La calificación final está compuesta por:

Evaluación	Porcentaje	Actividad	Porcentaje
Primer Parcial	20%	Tarea 1	15%
		Tarea 2	15%
		Evaluación Parcial 1	70%
Segundo Parcial	20%	Tarea 3	15%
		Tarea 4	15%
		Evaluación Parcial 2	70%
Final	53%	Tarea 5	10%
		Tarea 6	10%
		Proyecto final	40%
		<b>Examen Final</b>	<b>40%</b>
Semana i	7%		
<b>Total</b>	<b>100%</b>		

Las actividades mencionadas pueden ser individuales o grupales según indique el profesor.

En las actividades individuales es importante que realices una reflexión acerca de tu desempeño (autoevaluación) en la que puedas valorar la relación entre el esfuerzo que realizaste y la calificación que obtuviste.