



PROGRAMACIÓN DEL SÍLABO DE APRENDIZAJE

I. DATOS GENERALES

- | | | |
|------------------------------|---|---|
| 1. Nombre de la asignatura | : | Estructura de Datos |
| 2. Código del curso | : | B.2052 |
| 3. Año de estudios | : | Tercer Ciclo – I Semestre |
| 4. Cantidad de horas semanal | : | 06 (Teoría: 02 Práctica: 02 Laboratorio: 02) |
| 5. Créditos | : | 04 |
| 6. Docentes responsables | : | Ing. Hugo Manuel Barraza Vizcarra
hmbarrazav@gmail.com |
| 7. Jefe de prácticas | : | Ing. Edith Elizabeth Alfaro Gonzales
edithalg@hotmail.com |
| 8. Duración | : | 15 semanas |
| 9. Fecha de inicio | : | 18 de abril de 2016 |
| 10. Fecha de término | : | 29 de julio de 2016 |
| 11. Año académico | : | 2016 |

II. APOORTE DE LA ASIGNATURA AL PERFIL PROFESIONAL

Tiene como propósito brindar al futuro profesional de Ingeniería en Informática y Sistemas, conocimientos para desarrollar algoritmos de distintos tipos de datos con énfasis en las modernas teorías de programación. Presenta las herramientas necesarias para el análisis y evaluación de eficiencia de programas y las distintas alternativas que pueden aplicarse para resolver problemas complejos.

III. SUMILLA

El curso de Estructura de Datos corresponde al área de formación tecnológica siendo de carácter teórico – práctico. Se propone desarrollar la capacidad de análisis, diseño e implementación de algoritmos en el ordenador haciendo uso de un lenguaje de programación de propósito general para la estructuración de los datos en memoria. El desarrollo de la Asignatura comprende los aspectos de operaciones con estructura de datos tipo arreglos, registros, procesos recursivos, estructuras dinámicas tipo listas enlazadas y árbol para que finalmente el alumno construya aplicaciones computacionales utilizando alguna de estas estructuras.

IV. LOGRO DE LA ASIGNATURA

Al finalizar la asignatura, el alumno desarrolla aplicaciones computacionales utilizando datos estructurados que permita almacenar gran cantidad de datos en memoria, seleccionando la estructura de datos adecuada y aplicando las métricas de complejidad de los algoritmos.

V. PROGRAMACIÓN DE CONTENIDOS

PRIMERA UNIDAD DIDÁCTICA: ESTRUCTURAS FUNDAMENTALES TIPO ARREGLOS	
LOGRO: Al finalizar la primera unidad didáctica, el estudiante implementa procesos computacionales utilizando estructuras de datos estáticas del tipo arreglo unidimensional, determinando la complejidad de las estructuras en función al tiempo que tarda en acceder a los datos y validando en el ordenador los algoritmos generados.	
SEMANA	CONTENIDOS CONCEPTUALES
01	TEORÍA: Clasificación de las estructuras de datos. Estructuras de datos estáticas tipo Arreglos unidimensionales. Diseño de algoritmos para efectuar operaciones de entrada/salida. La clase arreglo. PRÁCTICA: Programación en el lenguaje de programación C++ de los algoritmos diseñados en las clases teóricas.
02	TEORÍA: Operaciones con arreglos: Inserción, modificación, eliminación. Operación sobre arreglos unidimensionales ordenados. Operación sobre arreglos unidimensionales desordenados. PRÁCTICA: Programar las operaciones de inserción, eliminación y modificación en arreglos unidimensionales

03	<p>TEORÍA: Combinación de arreglos y registros. Aplicación de arreglos bidimensionales en el álgebra lineal. Arreglos n-dimensionales.</p> <p>PRÁCTICA: Codificación de algoritmos usando arreglos bidimensionales aplicados al álgebra lineal y para aplicaciones de propósito general.</p>
----	--

SEGUNDA UNIDAD DIDÁCTICA: ESTRUCTURAS DE DATOS TIPO PILAS Y COLAS

LOGRO: Al finalizar la segunda unidad didáctica, el estudiante implementa aplicaciones computacionales utilizando estructuras de datos tipo Pila y Cola, diferenciando las características y propiedades de cada una de ellas, codificando en un lenguaje de programación soluciones computacionales y utilizando pilas y/o colas para almacenar datos.

SEMANA	CONTENIDOS CONCEPTUALES
04	<p>TEORÍA: Abstracción de datos tipo Pila. Implementación de Pilas usando estructuras estáticas. Tratamiento de expresiones aritméticas. La clase Pila.</p> <p>PRÁCTICA: Implementar en el ordenador los algoritmos diseñados para la manipulación de datos en la estructura tipo Pila.</p>
05	<p>TEORÍA: Abstracción de datos tipo Cola, Implementación de Colas usando estructuras estáticas. Colas circulares. Colas con prioridad. La clase Cola.</p> <p>PRÁCTICA: Implementar en el ordenador los algoritmos diseñados para la manipulación de datos en la estructura tipo Cola.</p>
06	PRIMER EXAMEN PARCIAL

TERCERA UNIDAD DIDÁCTICA: ESTRUCTURAS DE DATOS DINÁMICAS LINEALES

LOGRO: Al finalizar la tercera unidad didáctica, el estudiante elabora programas computacionales utilizando estructuras de datos dinámicas para contener la información, utilizando referencias a direcciones físicas de memoria del ordenador, evaluando el tiempo de respuesta en las operaciones de actualización de la estructura.

SEMANA	CONTENIDOS CONCEPTUALES
07	<p>TEORÍA: Tipos de datos abstractos representados mediante el uso de apuntadores de direcciones de memoria.</p> <p>PRÁCTICA: Implementación en el lenguaje C++ aplicaciones usando punteros para referenciar datos.</p>
08	<p>TEORÍA: Construcción de Listas enlazadas lineales. Diseño de algoritmos para las operaciones de actualización contenidos en las listas simplemente enlazadas:</p> <ul style="list-style-type: none"> • Creación • Inserción • Eliminación • Modificación <p>PRÁCTICA: Implementación y validación de los algoritmos diseñados en las clases teóricas.</p>
09	<p>TEORÍA: Construcción de Listas enlazadas lineales circulares. El nodo de fin de lista apunta al nodo de inicio o cabecera. Aplicar los apuntadores para definir listas doblemente enlazadas. Diseño de algoritmos para las operaciones de actualización contenidos en las listas doblemente enlazadas:</p> <ul style="list-style-type: none"> • Creación • Inserción • Eliminación • Modificación <p>PRÁCTICA: Implementación y validación de los algoritmos diseñados en las clases teóricas.</p>

CUARTA UNIDAD DIDÁCTICA: ESTRUCTURAS DE DATOS TIPO ÁRBOL

LOGRO: Al finalizar la cuarta unidad didáctica, el estudiante desarrolla aplicaciones computacionales utilizando estructuras de datos dinámicas no lineales, jerarquizando los datos contenidos en la estructura, efectuando operaciones de mantenimiento de la estructura, seleccionando el tipo de recorrido del árbol para el reporte de los datos que almacena.

SEMANA	CONTENIDOS CONCEPTUALES
10	TEORÍA: Estructuras no-lineales. Propiedades y características. Árboles generales de datos. Árboles binarios de datos. Árboles binarios completos. Representación de árboles binarios en memoria. Operaciones básicas: Creación, Recorrido. PRÁCTICA: Implementación de estructuras de datos no lineales en el computador.
11	TEORÍA: Árboles binarios de búsqueda (ABB). Representación de árboles generales como árboles binarios. PRÁCTICA: Programar en laboratorio los algoritmos para los ABB.
12	TEORÍA: Operaciones de inserción y eliminación de datos en los ABB: Inserción, eliminación. Eficiencia de la búsqueda en un árbol binario de búsqueda. ▪ Árboles Binarios de Búsqueda Balanceados: árboles AVL. Factor de equilibrio. PRÁCTICA: Implementación de las estructuras de datos tipo árboles AVL
13	TEORÍA: Operaciones con árboles AVL: ▪ Inserción, eliminación ▪ Rotaciones simples de nodos ▪ Rotaciones compuestas de nodos Análisis de eficiencia de un árbol AVL. Árboles autobalanceables: AA tree, Red and black tree, Splay tree, Radix tree. Grafos dirigidos y no dirigidos. PRÁCTICA: Análisis estructural de las estructuras indexadas.
14	EXPOSICIONES DE TRABAJOS DE INVESTIGACIÓN
15	SEGUNDO EXAMEN PARCIAL

VI. SISTEMA DE EVALUACIÓN

PROCEDIMIENTOS		PORCENTAJE	PONDERACIÓN
TEORÍA 60%	Exámenes parciales	50%	10
	Asistencia, evaluación continua, esfuerzo personal	10%	2
PRÁCTICA 40%	Trabajo de investigación	20%	4
	Implementación de algoritmos en laboratorio	20%	4

OBSERVACIÓN:

- La calificación será vigesimal (de 00 a 20) siendo la nota mínima de aprobación 11 (once).
- Solo para el promedio final se considerará 0.5 a favor del estudiante.
- Los alumnos que no se presenten a rendir sus evaluaciones en las fechas indicadas y definidas en clase tendrán la calificación 00. En caso la inasistencia sea debidamente justificada mediante los canales respectivos, según normas vigentes de la universidad, se reprogramará dicha evaluación.
- La evaluación de los estudiantes es permanente y el contenido de los exámenes es de todo el desarrollo del curso a la fecha de aplicación de la evaluación. La evaluación del estudiante es continua y no necesariamente requiere de aviso previo (excepto los exámenes parciales), siempre que las evaluaciones se realicen en horarios establecidos para el curso.
- El porcentaje de inasistencia a clases y/o laboratorio de más del 30% será causal de desaprobación del curso.

El promedio final del curso se obtiene de acuerdo a los criterios de evaluación que se señalaron anteriormente.

VII. ESTRATEGIAS METODOLÓGICAS

CLASES TEÓRICAS

- Método : Analítico y deductivo.
- Técnica : Exposición temática en aula propiciando el análisis y participación activa del alumno.
- Procedimiento : Planteamiento de problemas y la búsqueda de soluciones mediante el planteamiento de algoritmos computacionales. La complejidad de los problemas será progresiva, de acuerdo al avance del curso.

PRÁCTICAS DE LABORATORIO Y/O EN AULA

- Método : Analítico y deductivo.
- Técnica : Implementación de algoritmos mediante el uso de lenguajes de programación de alto nivel.

- c) Procedimiento : Planteamiento de problemas y la búsqueda de soluciones mediante el planteamiento de algoritmos computacionales. La complejidad de los problemas será progresiva, de acuerdo al avance del curso.

VIII. MATERIALES EDUCATIVOS Y OTROS RECURSOS DIDÁCTICOS

- Se utilizará material impreso y en formato digital.
- Pizarra acrílica, plumones, proyector multimedia y el computador.

IX. BIBLIOGRAFÍA

BIBLIOGRAFIA GENERAL

- JOYANES, L. (2006). Fundamentos de Programación: Algoritmos y Estructuras de datos (3ª edición). México: McGraw-Hill.
- KRUSE, R. (1996). Estructura de Datos y Diseño de Programas. México: Prentice-Hall Hispanoamericana.
- CAIRÓ, O. y GUARDATI, S. (2007). Estructura de Datos (3ª edición). México: McGraw-Hill.
- GUTIERREZ, J.M. (2006). Programación Avanzada. Alcalá – España: Universidad de Alcalá, Servicio de Publicaciones
- TENENBAUM A. Y AUGENSTEIN, M. (2004). Estructura de Datos en C. México: Prentice Hall Hispanoamericana

BIBLIOGRAFÍA ESPECÍFICA

- DROZDEK, A. (2005). Data Structures and Algorithms in C++ (2ª ed). Massachusetts: Grace Fujimoto
- HEILEMAN G. (1999). Estructura de Datos, Algoritmos y Programación Orientada a Objetos (2ª ed.). México: Mc Graw Hill
- LOOMIS, M. (1997). Estructura de Datos y Organización de Archivos. México: Prentice Hall Hispanoamericana
- MANBER, U. (2001). Introduction to Algorithms: A Creative Approach. New York: Addison-Wesley Iberoamericana.
- JOYANES L. (2006), Estructuras de datos en C++ (1ra ed). España: McGraw-Hill
- GUARDATI S. (2007). Estructura de datos orientada a objetos: Algoritmos con C++. México: Pearson Educación
- DALE N. (2007), Programación y resolución de problemas con C++. México: McGraw-Hill.
- CEBALLOS J. (2006), Enciclopedia del lenguaje C++, México: Alfaomega.
- SCHILDT, Herbert (2002). C++ Guía de referencia completa (4ª edición) Editorial Review.
- LIZA C. (2012), Algoritmos y su codificación en C++, (Volumen 3). Perú: Grupo Creadores.
- R. MARTINEZ, E. QUIROGA (2002), Estructura de Datos, Referencia práctica con orientación a objetos, Thomson Learning.
- LEWIS (2006), Estructura de datos con java, diseño de estructuras y algoritmos, Segunda edición, Editorial Pearson.