

SYLLABUS DE ALGORITMOS Y ESTRUCTURAS DE DATOS

I. DATOS GENERALES

Facultad: Ciencia, Tecnología y Ambiente			Departamento: Desarrollo Tecnológico	
Carrera: Ingeniería en Sistemas de Información			Nombre de la asignatura: Algoritmos y Estructuras de Datos	
Código de la asignatura: 270060			Créditos: 4	
Prerrequisito: 22054 Laboratorio de Programación I			Año y Cuatrimestre: 1 ^{er} . Año, Cuatrimestre 3	
HAD: 36	HAM: 0	HAV: 24	HAE: 60	Horas Totales: 120
Rasgos del perfil profesional que se desarrollan: <ol style="list-style-type: none"> Desarrolla software aplicando las teorías, principios, metodologías y mejores prácticas de la ingeniería de software, que aporten a resolver problemas organizacionales, demostrando comportamiento ético, con base en normas, en estándares y en recomendaciones nacionales e internacionales. Evalúa la calidad del proceso de desarrollo y de productos de software, aplicando modelos de evaluación, con el fin de garantizar la conformidad de estos con los estándares nacionales e internacionales 				Ejes Curriculares a los que pertenece la asignatura: <ul style="list-style-type: none"> Investigación Científico-Técnico
Programa de Investigación Formativa: Sí Competencia I Nivel 1: (a, b, c, d, e, f) Nivel 2: (c) Nivel 3: (a, b) Competencia II Nivel 1: (a)			Asignatura de Servicio Social: No	
Nombre del profesor: Prof. Ing. Jorge L. Morales R.			Tipo de contrato: Horario	
Grupo de clase: B009			Días y horario de clases: Miércoles de 7:00 a.m. a 9:30 a.m. (Lab J4) 2 horas en Modalidad Virtual	
Coordinador del Área: Prof. Lic. Armando J. López L.			Fecha de entrega: 24 de enero, 2019	

Revisado y Aprobado por: Prof. Lic. Armando J. López L.

II. OBJETIVOS DE LA ASIGNATURA

a) Generales:

Al concluir esta asignatura los/las estudiantes estarán en capacidad de:

1. Construir abstracciones eficientes de conjuntos de datos simples y complejos mediante el estudio de estructuras de datos reconocidas, así como los algoritmos asociados a éstas, para su implementación en programas basados en tales abstracciones utilizando un lenguaje de programación estructurado.
2. Diseñar algoritmos bien estructurados para el mejoramiento del diseño y del rendimiento de programas, basándose en el estudio y aplicación de estrategias algorítmicas reconocidas, sus ejemplares, y la implementación de buenas prácticas de programación.
3. Comprender con pensamiento analítico los problemas algorítmicos de ordenamiento y búsqueda, sus características y variantes, así como las soluciones existentes y sus ejemplares, a la luz del estudio de algoritmos de ordenación y de búsqueda reconocida y su implementación utilizando un lenguaje de programación estructurado.
4. Realizar con sentido crítico comparaciones de la eficiencia de algoritmos a fin de concluir sobre la mejor alternativa algorítmica ante un escenario problemático determinado, basándose en el análisis asintótico del comportamiento de algoritmos en cuanto al uso de los recursos de tiempo y espacio.

b) Específicos:

Al concluir la **primera unidad**, los estudiantes estarán en capacidad de:

- Realizar abstracciones de datos mediante su organización en estructuras de datos fundamentales y el estudio de su lógica organizativa y funcional.
- Gestionar la memoria utilizada por las estructuras de datos mediante su implementación como parte elemental de soluciones algorítmicas optimizadas.
- Implementar algoritmos asociados a las estructuras de datos fundamentales para la resolución de problemas organizativos de datos en la memoria utilizada por los programas.

Al concluir la **segunda unidad**, los estudiantes estarán en capacidad de:

- Aplicar buenas prácticas de programación al construir algoritmos, apoyándose de un proceso metodológico adecuado.
- Emplear estrategias algorítmicas para el diseño y la implementación de algoritmos computacionales.

Al concluir la **tercera unidad**, los estudiantes estarán en capacidad de:

- Explicar los conceptos de complejidad computacional y su impacto en el diseño y eficiencia de los algoritmos.
- Aplicar técnicas de análisis asintótico de algoritmos para medir eficiencia y eficacia en la construcción de soluciones algorítmicas.
- Analizar algoritmos fundamentales de ordenamiento y búsqueda mediante su implementación en contextos problemáticos particulares.

III. CONTRATO DIDÁCTICO

a) Durante las clases presenciales, los y las estudiantes se comprometen a:

- Asistencia y puntualidad obligatoria. Todo estudiante con 20% de inasistencias no justificadas perderá su derecho a calificación.
- Ser participativo en las clases.
- Entregar las evaluaciones en el tiempo establecido y las asignaciones en tiempo y forma, según lo acordado entre los estudiantes y el profesor.
- Conocer y respetar el reglamento de la universidad para lograr el ambiente de trabajo necesario para el desarrollo de las clases.
- Garantizar el buen estado de cualquier medio de almacenamiento para la entrega de asignaciones, exámenes o trabajo de curso.

b) Durante las clases virtuales, los y las estudiantes se comprometen a:

- Realizar las actividades académicas (tareas, evaluaciones, foros, consultas, etc.) únicamente mediante el Entorno Virtual de Aprendizaje (EVA), exceptuando aquellas para las cuales el docente indique el uso de una herramienta complementaria.
- Evitar salir del EVA durante la realización de actividades sincrónicas.
- Entregar las tareas y realizar las actividades virtuales en tiempo y forma, según el calendario establecido en este documento o según lo acordado entre los estudiantes y el profesor. El estudiante con 20% de incumplimiento de las actividades del curso, perderá su derecho a calificación final.
- No realizar fraude académico en las actividades académicas, individuales o grupales. En caso de cometer fraude, el/los estudiante(s) involucrado(s) está(n) sujeto(s) a pérdida de la calificación asociada.
- Respetar los derechos de autor en las tareas/ trabajos realizados, citando según la normativa APA 6ta. Edición.
- Conocer y respetar el reglamento académico estudiantil para lograr el ambiente de trabajo necesario para el desarrollo adecuado de las clases.
- Mantener una actitud respetuosa con sus compañeros(as) y docente.
- Evaluar constructivamente al tutor, con el propósito de mejorar el ejercicio de su docencia.

c) Durante las clases, el docente se compromete a:

- Desarrollar los temas planteados en el Syllabus.
- Retroalimentar al estudiante sobre su aprendizaje.
- Mantener una actitud respetuosa con sus estudiantes.
- Orientar de forma clara, coherente y oportuna las actividades a realizar, de acuerdo a lo consignado en las agendas de trabajo.
- Proporcionar los materiales de estudio que apoyen al estudiante en el aprendizaje por medios virtuales.
- Informar a los estudiantes el resultado de las evaluaciones y acumulado alcanzado. Retroalimentará una evaluación a más tardar 7 días después de realizada por los estudiantes.
- Brindar tutoría de la asignatura según el calendario establecido y con la calidad académica requerida, mediante Foro de Consulta General disponible en el EVA o a través de encuentros sincrónicos con las y los estudiantes.

IV. PLANIFICACIÓN DE LAS ACTIVIDADES

Las actividades por unidad se han planificado en correspondencia con las agendas de trabajo, organizadas de la siguiente manera:

Unidad	Horas Presenciales	Agendas Virtuales
I	12	4 (28 horas)
II	12	4 (28 horas)
III	12	2 (28 horas)

Unidad I: Estructuras de Datos			
Fecha	Temas, subtemas o ejes problematizadores	Actividades de Enseñanza y Aprendizaje	Materiales y recursos, guías de aprendizaje
Del 28 de enero al 3 de febrero	1. Revisión de las estructuras elementales 1.1. Arreglos unidimensionales 1.2. Registros y arreglos de registros 1.3. Cadenas de caracteres 2. Operaciones con cadenas de caracteres 2.1. Cálculo de longitud 2.2. Comparación 2.3. Concatenación 2.4. Extracción de subcadenas 2.5. Búsqueda 2.6. Inserción, eliminación, sustitución	1. Presentaciones <ul style="list-style-type: none"> Estructuras elementales: Arreglos, registros y cadenas caracteres 2. Lecturas <ul style="list-style-type: none"> Arrays (arreglos) unidimensionales: Los vectores Las cadenas de caracteres Registros 3. Cuestionario <ul style="list-style-type: none"> Cuestionario: Estructuras elementales 4. Prácticas <ul style="list-style-type: none"> Guía de Ejercicios #1: Operaciones con cadenas 	<ul style="list-style-type: none"> Syllabus de la asignatura Agenda Virtual # 1 Agenda Virtual # 2 Entorno Virtual de Aprendizaje Cuestionario: Estructuras elementales Guía de Ejercicios #1: Operaciones con cadenas Guía de Ejercicios #2: Operaciones con pilas y colas Libros de Texto Básicos <ul style="list-style-type: none"> Joyanes, L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos 4^{ta} Edición. Madrid; McGraw-Hill. Cairo, O., Guardati, S. (2012). Estructuras de Datos. 3^{ra} Edición. México; McGraw-Hill.
Del 4 al 10 de febrero	3. Arreglos 3.1. Arreglos unidimensionales paralelos 3.2. Arreglos bidimensionales 4. Arreglos multidimensionales 5. Estructuras estáticas lineales 5.1. Pilas 5.1.1. Inserción 5.1.2. Extracción 5.2. Colas 5.2.1. Inserción 5.2.2. Extracción	1. Presentaciones <ul style="list-style-type: none"> Arreglos Estructuras lineales (pilas y colas) 2. Lecturas <ul style="list-style-type: none"> Arrays multidimensionales Pilas Colas 3. Foros <ul style="list-style-type: none"> Debate sobre uso de Arreglos 4. Prácticas <ul style="list-style-type: none"> Guía de Ejercicios #2: Operaciones con pilas y colas 	

Unidad I: Estructuras de Datos			
Fecha	Temas, subtemas o ejes problematizadores	Actividades de Enseñanza y Aprendizaje	Materiales y recursos, guías de aprendizaje
Del 11 al 17 de febrero	7. Gestión dinámica de memoria en C/C++ 8. Recursión 9. Estructuras dinámicas lineales 9.1. Listas enlazadas 9.2. Listas doblemente enlazadas 9.3. Listas circulares 9.4. Doble cola	1. Presentaciones <ul style="list-style-type: none"> Gestión dinámica de memoria en C/C++ Recursión Estructuras dinámicas lineales 2. Lecturas <ul style="list-style-type: none"> Recursión Listas 3. Investigación <ul style="list-style-type: none"> Gestión dinámica de la memoria 4. Prácticas <ul style="list-style-type: none"> Guía de ejercicios #3: Listas circulares y doble cola 	<ul style="list-style-type: none"> Syllabus de la asignatura Agenda Virtual # 3 Agenda Virtual # 4 Entorno Virtual de Aprendizaje Cuestionario de la Unidad I Guía de Ejercicios #3: Listas circulares y doble cola Guía de Ejercicios #4: Árboles binarios de búsqueda y Grafos dirigidos Libros de Texto Básicos <ul style="list-style-type: none"> Joyanes, L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos 4^{ta} Edición. Madrid; McGraw-Hill.
Del 18 al 24 de febrero	10. Estructuras de datos no lineales 10.1. Árboles binarios 10.1.1. Conversión de un árbol general a un árbol binario 10.1.2. Recorrido de un árbol binario 10.2. Árbol binario de búsqueda 10.3. Árboles binarios completos 10.4. Árboles binarios balanceados 10.5. Grafos 10.5.1. Grafos dirigidos 10.5.2. Grafos no dirigidos	1. Presentaciones <ul style="list-style-type: none"> Estructuras dinámicas no lineales 2. Lecturas <ul style="list-style-type: none"> Árbol binario Grafos 3. Prácticas <ul style="list-style-type: none"> Guía de ejercicios #4: Árbol binario de búsqueda y Grafos dirigidos 4. Cuestionario <ul style="list-style-type: none"> Cuestionario de la Unidad I 	<ul style="list-style-type: none"> Cairo, O., Guardati, S. (2012). Estructuras de Datos. 3^{ra} Edición. México; McGraw-Hill.

Unidad II: Estrategias algorítmicas y buenas prácticas			
Fecha	Temas, subtemas o ejes problematizadores	Actividades de Enseñanza y Aprendizaje	Materiales y recursos, guías de aprendizaje
Del 25 de febrero al 24 de marzo	<ol style="list-style-type: none"> Diseño efectivo de algoritmos y programas <ol style="list-style-type: none"> Estilo de programación Estructura del programa <ol style="list-style-type: none"> Diseño descendente Diseño de programas Etapas de codificación Buenas prácticas de programación Etapas de pruebas Documentación de programas <ol style="list-style-type: none"> Documentación interna Documentación externa Estrategias algorítmicas <ol style="list-style-type: none"> Recursión Divide y vencerás Fuerza bruta Programación dinámica Algoritmos voraces Vuelta atrás Ramificación y acotamiento Emparejamiento de patrones y algoritmos de cadenas Algoritmos heurísticos Algoritmo de aproximación 	<ol style="list-style-type: none"> Presentaciones <ul style="list-style-type: none"> Diseño efectivo de algoritmos Diseño de programas Documentación de programas Estrategias algorítmicas Lecturas <ul style="list-style-type: none"> Escritura de algoritmos/programas Abstracción y resolución de problemas Estilo de programación La documentación Pruebas(testing) Cuestionario <ul style="list-style-type: none"> Diseño efectivo de algoritmos y programas Investigaciones <ul style="list-style-type: none"> Estrategias algorítmicas fundamentales Prácticas <ul style="list-style-type: none"> Guía de Ejercicios #5: Fuerza bruta Guía de Ejercicios #6: Divide y vencerás 	<ul style="list-style-type: none"> Syllabus de la asignatura Agenda Virtual # 5 Agenda Virtual # 6 Agenda Virtual # 7 Agenda Virtual # 8 Entorno Virtual de Aprendizaje Cuestionario: Diseño efectivo de algoritmos Guía de Ejercicios #5: Fuerza bruta Guía de Ejercicios #6: Divide y vencerás Libros de Texto Básicos <ul style="list-style-type: none"> Joyanes, L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos 4^{ta} Edición. Madrid; McGraw-Hill. Cairo, O., Guardati, S. (2012). Estructuras de Datos. 3^{ra} Edición. México; McGraw-Hill. Brassard, G., Bratley, P. Fundamentos de Algoritmia. 2^{da} Edición.

Unidad III: Análisis de la eficiencia de algoritmos			
Fecha	Temas, subtemas o ejes problematizadores	Actividades de Enseñanza y Aprendizaje	Materiales y recursos, guías de aprendizaje
Del 25 de marzo al 7 de abril	<ol style="list-style-type: none"> Fundamentos de algoritmia <ol style="list-style-type: none"> Problemas y ejemplares Complejidad computacional de los algoritmos Principios de invariancia Eficiencia de los algoritmos <ol style="list-style-type: none"> ¿Por qué se debe buscar? Operaciones elementales Notación asintótica de algoritmos <ol style="list-style-type: none"> Notación Θ (Theta) Notación Ω (Omega) Algoritmos de ordenación <ol style="list-style-type: none"> Ordenación por intercambio Ordenación por inserción Ordenación por selección Ordenación por Shell Ordenación rápida Ordenación por montículo Ordenación por intercalación Ordenación por fusión Análisis algorítmico <ol style="list-style-type: none"> Eficiencia y eficacia Compromisos espacio-tiempo Algoritmos de tiempo polinomial y de tiempo exponencial Análisis de algoritmos recursivos Análisis de las estructuras de control 	<ol style="list-style-type: none"> Presentaciones <ul style="list-style-type: none"> Problemas y ejemplares Complejidad computacional Principios de invariancia Notación asintótica de algoritmos Algoritmos de ordenación Análisis algorítmico Lecturas <ul style="list-style-type: none"> Problemas ejemplares ¿Por qué hay que buscar la eficiencia? Notación asintótica Eficiencia y eficacia Complejidad temporal y complejidad espacial Cuestionario <ul style="list-style-type: none"> Eficiencia de algoritmos y notación asintótica Prácticas <ul style="list-style-type: none"> Guía de ejercicios #7: Análisis de eficiencia de algoritmos. Guía de ejercicios #8: Algoritmos de ordenación 	<ul style="list-style-type: none"> Syllabus de la asignatura Entorno Virtual de Aprendizaje Agenda Virtual #9 Cuestionario: Eficiencia de algoritmos y notación asintótica Guía de Ejercicios #7: Análisis de eficiencia de algoritmos Guía de Ejercicios #8: Algoritmos de ordenación Libros de Texto Básicos <ul style="list-style-type: none"> Joyanes, L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos 4^a Edición. Madrid; McGraw-Hill. Cairo, O., Guardati, S. (2012). Estructuras de Datos. 3^{ra} Edición. México; McGraw-Hill. Brassard, G., Bratley, P. Fundamentos de Algoritmia. 2^{da} Edición.

Unidad III: Análisis de la eficiencia de algoritmos			
Fecha	Temas, subtemas o ejes problematizadores	Actividades de Enseñanza y Aprendizaje	Materiales y recursos, guías de aprendizaje
Del 8 al 25 de abril (la semana santa es del 15 al 21 de abril)	4. Análisis del “peor caso” 5. Análisis del “caso medio” 6. Medidas empíricas del rendimiento y comportamiento 7. Ordenación externa 7.1. Partición de archivos 7.1.1. Por clasificación interna 7.1.2. Por contenido 7.1.3. Por sustitución 7.1.4. Por secuencias 8. Algoritmos de búsqueda 8.1. Búsqueda secuencial 8.2. Búsqueda binaria 8.3. Búsqueda por transformación de claves 9. Ordenación por mezcla directa 10. Ordenación por mezcla natural 11. Ordenación por mezcla equilibrada 12. Ordenación por intercalación	1. Presentaciones <ul style="list-style-type: none"> Medidas de rendimiento y comportamiento Algoritmos de búsqueda Ordenación 2. Lecturas <ul style="list-style-type: none"> Análisis de “caso medio” y de “caso peor” Archivos ordenados Fusión de archivos Métodos de búsqueda 3. Cuestionario <ul style="list-style-type: none"> Cuestionario sobre medidas empíricas, “peor caso” y “caso medio” 4. Prácticas <ul style="list-style-type: none"> Guía Práctica #9: Algoritmos de búsqueda 	<ul style="list-style-type: none"> Syllabus de la asignatura Entorno Virtual de Aprendizaje Agenda Virtual #10 Cuestionario: Medidas empíricas, “peor caso” y “caso medio” Guía de Ejercicios #9: Algoritmos de búsqueda Libros de Texto Básicos <ul style="list-style-type: none"> Joyanes, L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos 4^{ta} Edición. Madrid; McGraw-Hill. Cairo, O., Guardati, S. (2012). Estructuras de Datos. 3^{ra} Edición. México; McGraw-Hill. Brassard, G., Bratley, P. Fundamentos de Algoritmos. 2^{da} Edición.

V. EVALUACIÓN DEL APRENDIZAJE

Actividades Evaluativas	Tipo de Evaluación	Puntaje	Fecha propuesta de realización
Cuestionario: Estructuras elementales	Formativa	-	2 de febrero
Guía de ejercicios #1	Formativa	-	3 de febrero
Foro: Debate sobre uso de arreglos	Formativa	-	Del 8 al 9 de febrero
Guía de ejercicios #2	Formativa	-	10 de febrero
Investigación: Gestión dinámica de memoria	Sumativa	5	15 de febrero
Guía de ejercicios #3	Sumativa	5	17 de febrero
Guía de ejercicios #4	Sumativa	5	22 de febrero
Cuestionario Unidad I	Sumativa	10	24 de febrero
Cuestionario: Diseño efectivo de algoritmos y programas	Sumativa	5	3 de marzo
Guía de ejercicios #5	Sumativa	5	10 de marzo
Investigación: Estrategias algorítmicas fundamentales	Sumativa	10	17 de marzo
Guía de ejercicios #6	Sumativa	5	24 de marzo
Cuestionario: Eficiencia de algoritmos y notación asintótica	Sumativa	5	30 de marzo
Guía de ejercicios #7	Sumativa	5	3 de abril
Guía de ejercicios #8	Sumativa	5	7 de abril
Cuestionario: medidas empíricas, “peor caso” y “caso medio”	Sumativa	5	13 de abril
Guía de ejercicios #9	Sumativa	5	23 de abril
Seguimiento y defensa de proyecto	Sumativa	30	Todo el cuatrimestre
Total de Calificación		100	

VI. BIBLIOGRAFÍA

1. Bibliografía Obligatoria

- a. Joyanes, L. (2008). Fundamentos de Programación. Algoritmos, estructura de datos y objetos 4^{ta} Edición. Madrid; McGraw-Hill.
- b. Cairo, O., Guardati, S. (2012). Estructuras de Datos. 3^{ra} Edición. México; McGraw-Hill.
- c. Brassard, G., Bratley, P. Fundamentos de Algoritmia. 2^{da} Edición.

2. Bibliografía Complementaria

- a. Aho, A., Hopcroft, J & Ullman, J. (1988). Estructura de Datos y algoritmos. México; PUBLICACION
- b. Baase. (2002). Algoritmos Computacionales. Introducción al análisis y diseño. México; Pearson Educación.
- c. Corona Nakamura, M. A. & Ancona Valdez, M. d. (2011). Diseño de Algoritmos y su codificación en Lenguaje C. México; McGraw-Hill/Interamericana Editores, S.A