



ESTRUCTURA DE DATOS I

2 CRÉDITOS TEÓRICOS – 2 PRÁCTICOS



A. INFORMACIÓN DEL PROFESOR

Nombre del profesor

Miguel Angel López Ortiz

Karen Elizabeth Liska Lima

e-mail

malopezort@correo.url.edu.gt

karenliska@gmail.com

Horario

Teoría: Presencial, lunes de 07:00 a 08:30 horas.

Laboratorio: Presencial, miércoles de 07:00 a 08:30 horas.



B. INFORMACIÓN GENERAL

Descripción

Este curso es el primero de dos cursos de Estructura de Datos, corresponde al área Profesional de Informática y forma parte del eje de Programación. El cual establece las bases necesarias para desarrollar la habilidad de resolver problemas a través de metodologías avanzadas para el análisis y desarrollo de algoritmos; aplicando distintas técnicas para la creación de programas eficientes y de calidad. Se le da continuidad a lo ya estudiado en Programación Avanzada, haciendo énfasis en algoritmos eficientes y abstracción de datos, se complementa el concepto de tipos de datos abstractos (TDA) y se da la introducción a los TDA no lineales y sus respectivas aplicaciones. Adicionalmente se prepara al alumno para realizar el análisis de algoritmos complejos, propios del área de las Ciencias de la Computación.

Modalidad

Presencial



C. COMPETENCIAS GENÉRICAS



El egresado landivariano se identifica por:

Pensamiento
lógico, reflexivo y
analógico

Pensamiento
crítico

Resolución de
problemas

Habilidades de
investigación

Uso de TIC y
gestión de la
información

Comunicación
efectiva, escrita y
oral

Comprensión
lectora

Compromiso
ético y
ciudadanía

Liderazgo
constructivo

Aprecio y respeto
por la diversidad e
interculturalidad

Creatividad

COMPETENCIAS ESPECÍFICAS (propias del curso)

Competencia 1: Analiza y calcula la complejidad de un algoritmo expresándola en la notación correcta.

Competencia 2: Comprende el funcionamiento de las distintas estructuras de datos lineales para resolver problemas.

Competencia 3: Comprende el funcionamiento de las distintas estructuras de datos no lineales para resolver problemas.

Competencia 4: Comprende el funcionamiento de las distintas estructuras de datos dispersas para resolver problemas.

Competencia 5: Construye aplicaciones funcionales utilizando el patrón de diseño Modelo Vista Controlador.

D. METODOLOGÍA

Este curso se desarrollará a través de los siguientes métodos de aprendizaje-enseñanza:

Aprendizaje invertido



«La exposición de saberes se realiza por medio de documentos, videos y otros materiales por parte del estudiante. El tiempo de sesión síncrona¹ se dedica a la discusión, resolución de problemas y actividades prácticas bajo la supervisión del profesor».

Aprendizaje basado en problemas (ABP)



«Metodología centrada en el aprendizaje, en la investigación y reflexión que siguen los alumnos para llegar a una solución ante un problema planteado. Desarrolla aprendizajes activos a través de la resolución de problemas y casos. Puede desarrollarse de manera síncrona o asíncrona»



E. PROGRAMACIÓN

COMPETENCIA 1

Analiza y calcula la complejidad de un algoritmo expresándola en la notación correcta.

Saber conceptual (contenido temático)

- 1.1 Complejidad de un algoritmo
- 1.2 Análisis asintótico

Saber procedimental (habilidades y destrezas)

- Reconoce la importancia de los pronósticos en el mundo empresarial y sus aplicaciones.
- Comprende los tipos de complejidad algorítmica por medio de un mapa mental o diagrama.
- Utiliza los conceptos para analizar y calcular la complejidad de algoritmos por medio de ejercicios individuales y grupales.

Saber actitudinal (conductas observables)

- Escucha con atención las explicaciones del curso.
- Participa activamente en las actividades de clase.
- Trabaja de forma colaborativa con sus compañeros.
- Realiza los trabajos asignados en tiempo.
- Presenta una solución completa y realizada con excelencia.

Indicadores de logro 1 (resultado): Analiza algoritmos, calcula su complejidad y la expresa en la notación correcta a través de resolución de ejercicios y prácticas.

COMPETENCIA 2

Comprende el funcionamiento de las distintas estructuras de datos lineales para resolver problemas.

Saber conceptual (contenido temático)

- 2.1. Plantillas y genéricos.
- 2.2. Delegados y predicados
- 2.3. TDA lineales

Saber procedimental (habilidades y destrezas)

- Comprende y utiliza correctamente las plantillas, genéricos, delegados y predicados por medio de resolución de problemas en laboratorio.
- Identifica los principales tipos de estructuras lineales, y es capaz de implementarlos en las prácticas de laboratorio.
- Construye un programa utilizando todos los conceptos aprendidos en un proyecto de aplicación.



Saber actitudinal (conductas observables)

- Escucha con atención las explicaciones del curso.
- Participa activamente en las actividades de clase.
- Trabaja de forma colaborativa con sus compañeros.
- Realiza los trabajos asignados en tiempo.
- Presenta una solución completa y realizada con excelencia.
- Trabaja de forma ordenada y comprensible apoyando el trabajo colaborativo.

Indicador de logro 2 (resultado):

- Implementa en código las distintas estructuras de datos lineales.
- Utiliza las distintas estructuras de datos lineales seleccionando la más adecuada para cada problema.

COMPETENCIA 3

Comprende el funcionamiento de las distintas estructuras de datos no lineales para resolver problemas.

Saber conceptual (contenido temático)

- 3.1. Árboles binarios
- 3.2. Árboles multcamino

Saber procedimental (habilidades y destrezas)

- Identifica los principales tipos de estructuras no lineales, y es capaz de implementarlos en las prácticas de laboratorio
- Construye un programa utilizando todos los conceptos aprendidos en un proyecto de aplicación.

Saber actitudinal (conductas observables)

- Escucha con atención las explicaciones del curso.
- Participa activamente en las actividades de clase.
- Trabaja de forma colaborativa con sus compañeros.
- Realiza los trabajos asignados en tiempo.
- Presenta una solución completa y realizada con excelencia.

Indicador de logro 3 (resultado):

- Implementa en código las distintas estructuras de datos no lineales.
- Implementa adecuadamente los distintos recorridos de cada estructura.
- Utiliza las distintas estructuras de datos no lineales seleccionando la más adecuada para cada problema.



COMPETENCIA 4

Comprende el funcionamiento de las distintas estructuras de datos dispersas para resolver problemas.

Saber conceptual (contenido temático)

- 4.1. Colas de prioridad
- 4.2 Tablas Hash
- 4.3 Arreglos asociativos
- 4.4 Otras estructuras

Saber procedimental (habilidades y destrezas)

- Identifica las estructuras de datos complejas, y es capaz de implementarlas en las prácticas de laboratorio y ejercicios en clase.
- Construye un programa utilizando todos los conceptos aprendidos en un proyecto de aplicación.

Saber actitudinal (conductas observables)

- Escucha con atención las explicaciones del curso.
- Participa activamente en las actividades de clase.
- Trabaja de forma colaborativa con sus compañeros.
- Realiza los trabajos asignados en tiempo.
- Presenta una solución completa y realizada con excelencia

Indicador de logro 4 (resultado):

- Implementa en código las distintas estructuras de datos dispersas.
- Utiliza las distintas estructuras de datos dispersas seleccionando la más adecuada para cada problema



COMPETENCIA 5

Construye aplicaciones funcionales utilizando el patrón de diseño Modelo Vista Controlador.

Saber conceptual (contenido temático)

- 5.1. Fundamentos de MVC
- 5.2. Fundamentos de aplicaciones web
- 5.3. Versionamiento de código
- 5.4. Patrón de diseño Singleton
- 5.5. Uso de POO en entorno web

Saber procedimental (habilidades y destrezas)

- Crea aplicaciones web funcionales.
- Controla los cambios de su aplicación utilizando un manejador de versiones.
- Diseña aplicaciones utilizando controladores, vistas y modelos de forma adecuada.
- Hace uso de la POO para conectar los componentes de las aplicaciones.
- Utiliza Singleton como simulación de almacenamiento de datos.

Saber actitudinal (conductas observables)

- Escucha con atención las explicaciones del curso.
- Participa activamente en las actividades de clase.
- Trabaja de forma colaborativa con sus compañeros.
- Realiza los trabajos asignados en tiempo.
- Presenta una solución completa y realizada con excelencia

Indicador de logro 5 (resultado):

- Presenta aplicaciones web MVC funcionales y construidas de forma adecuada.
- Maneja el versionamiento del código de sus aplicaciones en git.



F. EVALUACIÓN

a. Estrategias de evaluación sumativa

Estrategias	Puntaje
Pruebas Parciales	20
Tareas y exámenes cortos	10
Laboratorio	20
Proyecto de aplicación	20
Examen final	30
TOTAL	100

Es requisito indispensable aprobar el Laboratorio con una ponderación mínima del 65% (prácticas y proyecto de aplicación), para aprobar el curso. Se debe contar con un mínimo del 75% de asistencia a clases (teoría y laboratorio) para tener derecho a concluir el curso.

b. Estrategias de evaluación formativa

Técnicas formativas	Procedimiento
Retroalimentación	Se proporciona la resolución de todos los problemas de las hojas de trabajo y exámenes realizados.
Diálogo socrático	Preguntas y respuestas orales a ejemplos y problemas que se realizarán lo largo de la secuencia de aprendizaje.
Padlet colaborativo	Se utiliza en la actividad de contextualización y presentación del curso
Exámenes cortos	Problemas de aplicación del tema seleccionado.
Trabajos en pequeños grupos para resolver dudas	Hojas de trabajo que se resuelven de forma colaborativa entre estudiantes.
Citas individuales	Tutorías de retroalimentación solicitadas por el estudiante, por medios electrónicos



G. CALENDARIO DE REFERENCIA POR TEMAS

Fecha	Tema	Actividad de evaluación
Semana 1	Unidad 1: Complejidad de un algoritmo	Presentación de curso y clases magistrales
Semana 2	Unidad 1: Análisis asintótico	Análisis de casos y resolución de problemas
Semana 3	Unidad 1: Ejercicios prácticos	Análisis de casos, lluvia de ideas y resolución de problemas
Semana 4	Unidad 2: Programación genérica, plantillas y delegados	Clase magistral y examen corto
Semana 5	Unidad 2: Ejercicios prácticos	Análisis de casos y resolución de problemas
Semana 6	Unidad 3: Árboles binarios	Clase magistral
Semana 7	Unidad 3: Ejercicios prácticos	Análisis de casos y resolución de problemas
Semana 8	Unidades 1, 2, 3: Examen parcial	Examen parcial
Semana 9	Unidad 3: Árboles AVL	Clase magistral y examen corto
Semana 10	Unidad 4: Colas de prioridad	Clase magistral
Semana 11	Unidad 4: Tablas Hash	Clase magistral y examen corto
Semana 12	Unidad 4: Heapsort	Clase magistral, análisis de casos y resolución de problemas
Semana 13	Unidad 5: Árboles multcamino	Clase magistral
Semana 14	Unidad 5: Ejercicios prácticos	Clase magistral, examen corto, análisis de casos y resolución de problemas
Semana 15	Examen final	Examen final de curso

H. REFERENCIAS BIBLIOGRÁFICAS

- JOYANES AGUILAR, LUIS. **Fundamentos de Programación: Algoritmos, estructuras de datos y objetos**. (5ra. edición). Mc-Graw Hill. España, 2020.
- ECK, DAVID (2014). **Introduction to Programming Using Java**. Hobart and William Smith Colleges. 7ma Edición. <http://math.hws.edu/javanotes/index.html>
- BLELLOCH, GUY (2016). **Algorithm Design: Parallel and Sequential**. Carnegie Mellon University. 1ra Edición. <http://www.parallel-algorithms-book.com>
- STANDISH, THOMAS. Data Structures, Algorithms, and Software principles in C. Addison-Wesley. U.S.A. 1994.



Ing. Adolfo Galán

Director

Departamento de Ingeniería en
Informática y Sistemas
Facultad de Ingeniería