

Programa del curso CE2103

## **Algoritmos y Estructuras de Datos II**

Área Ingeniería en Computadores  
Licenciatura Ingeniería en Computadores

[Última revisión de la plantilla: 31 de agosto del 2017]

## 1 Datos generales

Nombre del curso:	Algoritmos y Estructuras de Datos II
Código:	CE2103
Tipo de curso:	Teórico – Práctico
Electivo o no:	No
Nº de créditos:	4
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	8
Áreas curriculares	Ciencias de Ingeniería (40%) Diseño en Ingeniería (60%)
Ubicación en el plan de estudios:	Curso de 3 <sup>er</sup> semestre de la carrera de Ingeniería en Computadores
Requisitos:	CE-1103: Algoritmos y Estructuras de Datos I EM-1601: Matemática Discreta
Co-requisitos:	No
El curso es requisito de:	CE-3104: Lenguajes, compiladores e Intérpretes EL-3307: Diseño Lógico
Asistencia:	Obligatorio

Suficiencia:	No
Posibilidad de reconocimiento:	Sí
Vigencia del programa:	II Semestre 2022

---

## 2 Descripción general

Este curso complementa el curso de Algoritmos y Estructuras de Datos I, ampliando las estructuras de datos) y los algoritmos relacionados con ellas. Hace énfasis en el diseño de algoritmos y el análisis de estos, para determinar cuáles son más eficientes que otros.

Los atributos de acreditación relacionados a los graduados y a este curso son los siguientes:

- ◆ **Análisis de problemas (Inicial):** Capacidad para identificar la naturaleza de los problemas de ingeniería de acuerdo con su campo de estudio.
- ◆ **Investigación (Inicial):** Capacidad para formular propuestas de investigación de problemas complejos de acuerdo con la metodología establecida.
- ◆ **Trabajo individual y en equipo (Medio):** Capacidad para colaborar de forma activa en equipos de trabajo para la realización de las labores asignadas.

Este curso es de vital importancia para el resto de los cursos que se imparten en la carrera ya que sienta las bases de pensamiento lógico y diseño lo cual es fundamental para comprender conceptos complejos que se estudiarán más adelante.

El curso es adaptable a necesidades educativas especiales que puedan tener los estudiantes.

La metodología del curso busca además reforzar ciertas habilidades no técnicas como el trabajo en equipo y la creatividad, así como los valores de responsabilidad, respeto y tolerancia. El desarrollo de estas habilidades y valores permitirá una ejecución óptima de las diferentes actividades de manera individual y grupal.

### 3 Objetivos

#### General:

Profundizar los conocimientos relacionados con algoritmos y estructuras de datos, apoyándose en el uso de técnicas de diseño y medición de los mismos, desarrollando a su vez habilidades de comunicación y trabajo en equipo.

Objetivos específicos	Atributos correspondientes
Mejorar el diseño y modelado de estructuras de datos en aplicaciones mediante el uso de buenas prácticas.	AP(I), IN(I), TE(M)
Implementar algoritmos para la manipulación de texto.	AP(I), IN(I), TE(M)
Medir la eficiencia de algoritmos.	IN(I), TE(M)
Desarrollar diferentes estrategias para el diseño de algoritmos.	AP(I), IN(I), TE(M)
Implementar diferentes técnicas para el manejo eficiente de la memoria secundaria.	AP(I), IN(I), TE(M)
Implementar el diseño y modelado de estructuras de datos y aplicaciones.	AP(I), IN(I), TE(M)
Aprender conceptos introductorios de sistemas de archivos y sistemas de bases de datos.	IN(I)

Aplicar patrones de diseño en la solución de un problema.	AP(I), IN(I), TE(M)
Desarrollar habilidades no técnicas como la comunicación, el trabajo en equipo y la creatividad, así como los valores de responsabilidad, respeto y tolerancia	TE(M)

#### 4 Contenidos

- Introducción a la administración de memoria **(2.5 semanas)**
  - ◆ Referencias en memoria
  - ◆ Declaración y uso eficiente de variables.
  - ◆ Parámetros por referencia y por valor.
  - ◆ Pila de llamadas
  - ◆ Técnicas de la administración de memoria en el heap.
  - ◆ Aritmética de direcciones.
- Análisis de algoritmos **(2 semanas)**
  - ◆ Medición empírica
  - ◆ Medición simulada
  - ◆ Medición Analítica
  - ◆ Notaciones más comunes (O grade, o pequeña  $\Omega$ )
- Algoritmos de Búsqueda **(1 semana)**
  - ◆ Búsqueda Secuencial
  - ◆ Búsqueda Binaria
  - ◆ Búsqueda por Interpolación
  - ◆ Hashing
  - ◆ Pathfinding

→ Patrones de Diseño, UML y calidad de código (**2 semanas**)

- ◆ Patrones constructivos
- ◆ Patrones estructurales
- ◆ Patrones de comportamiento
- ◆ Buenas prácticas de escritura de código.
- ◆ Casos de pruebas
- ◆ Tipos de pruebas
- ◆ Pruebas unitarias

→ Diseño de algoritmos (**3.5 semanas**)

- ◆ Ávidos
- ◆ Divide y vencerás
- ◆ Programación dinámica
- ◆ Backtracking
- ◆ Algoritmos probabilísticos
- ◆ Algoritmos genéticos
- ◆ Algoritmos heurísticos
- ◆ Bitvectors

→ Estructuras de almacenamiento externo (**2.5 semanas**)

- ◆ Estructuras jerárquicas (Árboles B y R)
- ◆ Sistemas de archivos
- ◆ Operaciones sobre archivos
- ◆ Archivos secuenciales
- ◆ Archivos relativos
- ◆ Archivos secuenciales indexados
- ◆ Archivos multillave
- ◆ Soluciones híbridas
- ◆ Cache de datos
- ◆ Algoritmos de ordenamiento externo

→ Algoritmos de compresión (**1.5 semanas**)

- ◆ Sin pérdida de datos
- ◆ Con pérdida de datos
- ◆ Códigos de huffman.
- ◆ Codificación aritmética.
- ◆ LZW

→ Estructuras básicas de bases de datos (1 semana)

- ◆ Introducción a los motores de bases de datos.
- ◆ Definición de tablas e índices.
- ◆ Operaciones básicas de SQL
- ◆ Conceptos avanzados de bases de datos.
- ◆ NOSQL

5 Metodología de  
enseñanza y aprendizaje

Se emplearán técnicas de clases magistrales por parte del profesor, en donde se desarrollarán los aspectos teóricos y prácticos más relevantes de los diferentes temas. Además, se combinarán con una alta participación por parte de los estudiantes durante el transcurso de las lecciones, por medio de llamadas orales, respuestas a casos en la pizarra y de trabajos individuales y en grupo.

Se presupone que el alumno profundiza los temas abordados en la clase en las lecturas recomendadas por el profesor y que el estudiante será responsable de desarrollar los diferentes entregables que se asignen en el curso.

El profesor asumirá el papel de facilitador y el estudiante tendrá la mayor responsabilidad de su progreso.

Las herramientas básicas para poder enfrentar este curso son:

1. Una computadora con mínimo 1GB de RAM y CPU Pentium 4.
2. Sistema Operativo GNU Linux (se recomienda <http://www.ubuntu.com/>)
3. GNU g++
4. Un editor de código fuente (se recomienda C Lion)

Si un estudiante no cuenta con estos recursos puede hacer uso de los diferentes laboratorios disponibles en el campus central del Instituto Tecnológico de Costa Rica.

## 6 Evaluación

Pruebas cortas	<b>10%</b>
Tareas extraclase	<b>10%</b>
Actividades en clase	<b>5%</b>
Exámenes	<b>15%</b>
Proyectos programados	<b>60%</b>
<b>TOTAL</b>	<b>100%</b>

La evaluación del curso presenta un balance entre el trabajo individual y en equipo de los estudiantes.

Las pruebas cortas se efectuarán en el momento de la clase que el profesor considere más apropiado. Por ningún motivo se realizarán reposiciones de pruebas cortas, excepto que exista una justificación válida con su respectiva evidencia.

Se asignan tareas extraclase que pueden estar relacionadas con análisis, diseño, programación, investigación y presentaciones orales. Se establecerán las reglas para cada una cuando se asignen. Estas tareas extraclase deberán ser desarrolladas por los estudiantes fuera del horario normal de clases.

El rubro de Actividades en Clase corresponde a actividades grupales que estén relacionados con aprendizaje significativo. Estas actividades pueden involucrar talleres relacionados con temas vistos en el curso o con tecnologías para el desarrollo de software.

Los exámenes cubrirán los contenidos del curso, y más que una comprobación de lectura o de memorización de contenidos, intentarán



determinar si el estudiante desarrolló los conocimientos y el criterio para aplicar los temas cubiertos. No existe la posibilidad de que el estudiante se exima de los exámenes.

Los proyectos programados se podrán realizar en grupos o de manera individual. En la especificación de cada proyecto se establecerán estos lineamientos. Las evaluaciones programadas deberán tener una adecuada documentación y los resultados deberán ser presentados al profesor en una defensa preparada adecuadamente. La defensa es obligatoria para la revisión del proyecto programado, de no ser así, no se procederá a calificar el mismo. Las normas para la documentación se entregarán oportunamente para cada proyecto.

Para el cálculo del porcentaje de pruebas cortas, tareas extraclase y actividades en clase, se corresponderá a calcular la proporción de todas las calificaciones obtenidas, donde cada entregable tendrá un peso idéntico para el cálculo de dicho rubro. Para los proyectos programados no necesariamente aplicará la misma regla. En el caso de las tareas extraclase, algunas podrían dividirse en pequeñas tareas que al final sumarán el valor de una tarea extraclase. Cuando esto suceda se avisará con anticipación a los estudiantes.

En caso de que se detecte un plagio o intento de fraude en cualquier trabajo, asignación o evaluación por parte de un estudiante, se procederá a anular el mismo y se enviará una carta al expediente del estudiante.

El orden, precisión, concisión, ortografía y redacción será evaluado en todos los documentos presentados. El profesor está en la libertad de revisar o no los documentos que no cumplan con dicho rubro.

El cronograma del curso puede ser visualizado en el siguiente enlace:

[Link al cronograma](#)

Este cronograma podrá sufrir cambios conforme avance el semestre y los mismos serán coordinados con anticipación con los estudiantes. El cronograma incluye las fechas para los grupos que se imparten martes/jueves y el de miércoles/viernes. Las fechas de los entregables corresponden al horario de cada uno de los grupos.

## 7 Bibliografía

### Obligatoria

- Deitel y Deitel, (2012). Cómo programar en Java. 9th ed. México: Pearson Educación.
- Drozdek, Adam, (2013). Data Structures and Algorithms in Java. 4th ed. United States: Cengage Learning Asia.

### Complementaria

- Weiss, M.A., "Data Structures & Algorithm. Analisis in JAVA", 1 edition, Addison Wesley Longman, Harlow, Inglaterra, 1999.
  - Aho, A.V., Hopcroft, J.E., Ullman, J.D., "Estructuras de datos y algoritmos", versión en español de la I edición, Addison-Wesley Iberoamericana, Buenos Aires, Argentina, 1988.
  - Joyanes, Luis y Zahonero Ignacio. Estructuras de Datos: Algoritmos, abstracción y objetos. Mc Graw Hill. 1998.
- 

## 8 Cuerpo docente

Ing. Luis Diego Noguera Mena, MGP.

Egresado del Instituto Tecnológico de Costa Rica.

### Coordinador del curso

Ha impartido el curso de CE1103 - Algoritmos y Estructuras de Datos I, CE2103 - Algoritmos y Estructuras de Datos II, CE3101 - Bases de Datos y CE4101 - Especificación y Diseño de Software en el Área Académica Ingeniería en Computadores, desde el 2016. En el ámbito profesional ha desempeñado los puestos de Desarrollador, Líder Técnico y Líder de equipo, siendo el responsable de los entregables de un equipo de desarrollo de aplicaciones web. Ha laborado en empresas como DataStreamLabs, HP, HPE y DXC Technology.

### Contacto

Email: lnoguera@tec.ac.cr

---

Célular: 8823-6765 (WhatsApp, Telegram)

Tel. Oficina: 2550-2566

Oficina: Edificio F2, Oficina 10

---

Ing. Isaac Ramírez, M.SI.

Egresado del instituto tecnológico de Costa Rica

Ha impartido el curso de Algoritmos y Estructuras de Datos I y II y Bases de Datos en el Área Académica Ingeniería en Computadores desde el 2014. Desde el 2008 ha ejercido en el área de computación en distintos roles como Programador, Ingeniero de Software y Supervisor de Desarrollo de Software. Actualmente se desempeña como Ingeniero de Software en Microsoft.

*Contacto*

asistente.isaac.itcr@gmail.com

Teléfono: 83237466 (WhatsApp, Telegram)

---

Ing. Leonardo Araya Martínez, M.Sc.

Egresado de Ingeniería en Computadores del Tecnológico de Costa Rica.

Ha impartido cursos relacionados a algoritmos y estructuras de datos, así como cursos de electrónica. También ha tenido a cargo la asesoría de varios trabajos finales de graduación. Todo dentro del Área Académica de Ingeniería en Computadores y la Escuela de Computación, desde el 2019.

Actualmente trabaja como diseñador de SoC en el departamento de Custom Compute de Intel.

**Contacto**

Email: learaya@itcr.ac.cr

Teléfono: 89699250 (Telegram (preferible), WhatsApp)

## **Consultas**

Horario y lugar de consulta a convenir con cada profesor.

## **Tutorías**

Horario y lugar de tutorías se indicará una vez iniciado el curso lectivo debido a que se debe esperar si el curso contará con tutoría. Si así fuera se debe definir horarios entre tutores y estudiantes.

### 9 Instrucciones generales

## **Medios de comunicación**

Cada profesor establecerá los medios de comunicación con los estudiantes y los medios para hacer entrega de las evaluaciones y para la entrega de materiales.

## **Entregables**

Cualquier documento que sea solicitado como entregable debe ser en formato PDF. En caso de ser necesario otro formato será indicado por el profesor.

Cada proyecto establece las reglas de entrega.

## **Sobre el proceso de acreditación de la carrera**

Dado que la carrera Ingeniería en Computadores está en proceso de reacreditación, el profesor podrá dejarse algunas copias de los entregables y quices.

## **Otros**

Está terminantemente prohibido realizar grabaciones de las lecciones sin autorización del profesor.