

FACULTAD DE INGENIERÍA, DISEÑO Y CIENCIAS APLICADAS / ESCUELA DE TECNOLOGÍA, DISEÑO E INNOVACIÓN***Información de la asignatura***

Nombre de la asignatura	Algoritmos y programación 2
Código de la asignatura	09704
Período académico	2023-II
Intensidad horaria¹	5 horas / 3 créditos
Intensidad semanal¹	5 horas /Semana
Créditos¹	3
Apoyo a Currículo Central	N/A
Profesor(es)	Domiciano Rincón / Juan David Bolaños / Nicolás Salazar

Introducción o presentación general del curso

En este curso se busca que cada estudiante tenga los conocimientos en el área de programación que le permitan implementar soluciones algorítmicas para resolver problemas de nivel intermedio en complejidad. Siendo éste el tercer curso del bloque de formación en algoritmos, se introducen nuevos conceptos para modelar las entidades de la solución y técnicas para implementar algoritmos que resuelvan problemas de una complejidad mayor a la que el estudiante ha tenido hasta el momento. Asimismo, se estudian conceptos fundamentales del diseño de la experiencia de usuario. Este curso pertenece a un bloque de conocimiento incremental, por lo que se supone que el estudiante cuenta con las competencias adquiridas en el curso previo de Algoritmos y Programación I.

Formación en competencias

- **SO-1. Solución de problemas:** Identificar, formular y resolver problemas complejos de ingeniería aplicando pensamiento crítico y principios de las ciencias, las matemáticas, la ingeniería y, en particular, de las Ciencias de la Computación y de la Ingeniería de Software.
- **SO-2. Diseño de ingeniería:** Diseñar soluciones y procesos basados en software que satisfagan necesidades específicas y generen valor a sus usuarios, considerando la salud pública, la seguridad y el bienestar de las personas, así como factores globales, culturales, sociales, ambientales y económicos.
- **SO-5. Trabajar de manera efectiva en equipos,** cuyos miembros brinden liderazgo de manera colectiva, creen entornos colaborativos e inclusivos, establezcan metas, planifiquen tareas y logren objetivos, mientras se adaptan a situaciones cambiantes.

¹ La intensidad horaria, intensidad semanal y créditos, deben estar alineados con la “Política de definición de créditos académicos de la Universidad Icesi” – Resolución N°. 80 de Junta Directiva del 22 de septiembre de 2019.

Objetivo general (meta de aprendizaje)

Resolver problemas combinando las ventajas de utilizar diferentes estrategias, enfoques y buenas prácticas de programación (orientada a objetos, concurrencia, recursividad y codificación segura) como parte de un proceso de desarrollo de la solución, consciente de la importancia de la gestión de recursos computacionales y del diseño de la experiencia de usuario, con una mayor capacidad de respuesta técnica a las necesidades concretas del contexto del problema.

Objetivos terminales (resultados de aprendizajes de la asignatura)

Competencia Curso	Competencia Egreso*	RA \equiv PI*
(OT1) Validar y controlar errores en los programas mediante el uso de pruebas unitarias y excepciones.	SO2-Diseño de ingeniería	SO2-PI1 Diseño detallado.
(OT2) Evaluar algoritmos de búsqueda y ordenamiento clásicos en estructuras de datos lineales y no lineales.	SO1-Solución de problemas	SO1-PI2 Solución de problemas (ingeniería y ciencias)
(OT3) Implementar soluciones a problemas que requieran el uso de listas enlazadas, árboles y recursión.	SO1-Solución de problemas	SO1-PI2 Solución de problemas (ingeniería y ciencias)
	SO2- Diseño de ingeniería	SO2-PI1 Diseño detallado.
(OT4) Desarrollar y desplegar programas que utilicen interfaces gráficas de usuario con persistencia de la información, incorporando buenas prácticas, conceptos de codificación segura y elementos básicos de gestión de la configuración.	SO1-Solución de problemas	SO1-PI2 Solución de problemas (ingeniería y ciencias)
	SO2- Diseño de ingeniería	SO2-PI1 Diseño detallado.
	SO5-Trabajo en equipo	SO5-PI1 Formar equipos efectivos.
(OT5) Construir programas bajo el paradigma de programación orientado a objetos concurrente, gestionando el acceso sincronizado a recursos compartidos en un nivel básico	SO-1-Solución de problemas	SO1-PI2 Solución de problemas (ingeniería y ciencias)
	SO2- Diseño de ingeniería	SO2-PI1 Diseño detallado.
	SO5-Trabajo en equipo	SO5-PI1 Formar equipos efectivos.

*NOTA: La descripción completa de los SOs y de los PIs puede consultarse haciendo click en los siguientes enlaces: [Student Outcomes](#) y [Performance Indicators](#)

Unidades de aprendizaje

Al finalizar la unidad, los estudiantes serán capaces de

Unidad 1 – Estructuras y Algoritmos Recursivos (3 semanas)

- Emplear el concepto de recursividad como una alternativa a la estructura de control iterativa.
- Aplicar la computación recursiva en la solución de problemas de naturaleza inherentemente autocontenida.
- Utilizar árboles binarios de búsqueda para representar grupos de objetos que mantienen entre ellos una relación de orden.
- Escribir algoritmos recursivos para manipular estructuras de información recursivas y explicar las ventajas que, en este caso, estos algoritmos tienen sobre los algoritmos iterativos.

Unidad 2 – Estructuras Lineales Enlazadas (3 semanas)

- Utilizar estructuras enlazadas de objetos para modelar grupos de atributos no primitivos de tamaño flexible.
- Escribir los algoritmos necesarios para manipular estructuras lineales que almacenan sus elementos enlazándolos entre ellos.
- Utilizar herramientas de diseño para la construcción de diagramas y la generación de código a partir de éstos.

Unidad 3 – Pruebas Automáticas y Tipos de excepción (2 semanas)

- Reconocer el mecanismo de manejo de excepciones señalando las implicaciones de la propagación versus el control.
- Usar e implementar distintos tipos de excepción como parte de un programa, de manera que sea posible clasificar los tipos de error que se pueden presentar y asociarles en el programa distintas maneras de recuperarse ante el problema.
- Diseñar pruebas unitarias automáticas que permitan validar el adecuado funcionamiento de las operaciones del sistema desarrolladas para soportar los requerimientos funcionales.
- Desarrollar las clases y los métodos necesarios para implementar las pruebas unitarias automáticas, que ayudan a comprobar el correcto funcionamiento de un programa.

Unidad 4 – Persistencia, Manejo de Archivos de Texto y Algoritmos de Ordenamiento y Búsqueda (3 semanas)

- Manipular archivos de texto para implementar requerimientos del cliente relacionados con persistencia
- Leer entradas e imprimir salidas de programas que interactúan directamente con archivos de texto y no con usuarios finales.
- Hacer persistir el estado del modelo de solución del problema durante la ejecución de un programa y restaurarlo cuando se requiera usando la técnica de serialización.
- Implementar algoritmos clásicos de ordenamiento de datos en estructuras de datos lineales y aplicarlos en la solución de un problema.
- Implementar algoritmos clásicos de búsqueda de información en estructuras de datos lineales y aplicarlos en la solución de un problema.
- Hacer uso de las interfaces Comparable y Comparator para definir relaciones de orden total sobre objetos.
- Calcular el tiempo de ejecución de un algoritmo por medio de las operaciones de tiempo del sistema
- Implementar métodos que permitan generar muestras con datos aleatorios.

Unidad 5 – Construcción de la interfaz gráfica (3 semanas)

- Utilizar una arquitectura de tres capas para el desarrollo de un programa de computador, repartiendo de manera adecuada las responsabilidades entre la interfaz de usuario, el control de la interfaz y el modelo. El estudiante deberá poder explicar la importancia de mantener separadas las clases de estos tres dominios.
- Construir las clases que implementan una interfaz de usuario.

- Aplicar la técnica de descomposición de requerimientos para cumplir con la funcionalidad de un programa de computador.

Unidad 6 – Concurrencia y Dibujo Básico en 2D (2 semanas)

- Desarrollar un programa que maneje concurrencia, de manera que sea posible que ejecute más de una parte del programa de manera simultánea, utilizando hilos de ejecución (threads).
- Construir interfaces de usuario que incluyan gráficas en 2 dimensiones como una alternativa en la presentación de información al usuario.

Metodologías de aprendizajes

Para los estudiantes

De acuerdo con la metodología de aprendizaje activo de la universidad Icesi, los estudiantes deben preparar, antes de la clase, los temas asignados en la programación del curso. Esto es:

Leer y analizar el material asignado para la sección de clase. Si no se ha asignado material, entonces investigar sobre los temas acordados en la planeación.

Utilizar estrategias de estudio (mapas conceptuales, mapas mentales, resúmenes, etc.) que sean efectivas y que sirvan como refuerzo después del proceso de lectura.

Contestar las preguntas que contiene el material, así como las preguntas adicionales que el profesor entregue.

Resolver los ejercicios propuestos en el material, así como los ejercicios adicionales que se le entreguen.

Formular preguntas que requieran ser resueltas durante la clase.

Durante la clase, el estudiante deberá:

Plantear las dudas que le quedaron durante el proceso de estudio del tema a tratar.

Participar en las actividades de revisión y consolidación de conceptos que proponga el profesor.

Trabajar en la solución de los problemas de aplicación que se propongan.

Después de la clase

Establecer las relaciones entre los temas tratados en la clase y el conocimiento previamente adquirido.

Resolver los ejercicios de aplicación del tema, que tienen un nivel de complejidad mayor al de los ejercicios que resolvió previamente.

Para el desarrollo del curso

Este curso cuenta semanalmente con dos tipos espacios que son utilizados de la siguiente forma: Sesiones teórico/prácticas: los estudiantes y el profesor se encontrarán en este espacio, en dos sesiones de una hora y media (una hora, 30 minutos) en las cuales se llevará a cabo la discusión de los diferentes temas y la realización de ejercicios que permitan ponerlos en práctica.

Sesiones principalmente prácticas: los estudiantes, el profesor y el monitor del curso comparten un espacio en el cual se llevan a cabo la solución de problemas propuestos y su implementación en el lenguaje Java. Para este componente se han destinado dos (2) horas por semana. Este curso utilizará Java como lenguaje de programación y IntelliJ como entorno de desarrollo. No obstante, los objetivos de aprendizaje son independientes del lenguaje y el entorno de programación seleccionado.

Evaluación de aprendizajes

Mecanismo o actividad evaluativa	Porcentaje de la nota final	Relación con objetivo terminales – resultado de aprendizaje del curso	Relación con el resultado de aprendizaje de la competencia de egreso = relación con el Indicador de desempeño del student outcome
Tarea Integradora 1	25 %	OT3	SO1-PI2; SO2-PI1
Tarea Integradora 2	25 %	OT1, OT3	SO1-PI2; SO2-PI1; SO5-PI1
Tarea Integradora 3	25 %	OT4, OT5	SO1-PI2; SO2-PI1; SO5-PI1
Seguimientos	25 %	OT1, OT2, OT3, OT4, OT5	SO1-PI2; SO2-PI1; SO5-PI1

Medios Educativos

No es necesario comprar ningún libro para el curso. Todo el material necesario para el curso será suministrado a través de Intu, como los materiales bibliográficos, guías de clases, videos, blogs, cursos en línea y otros.

Algunos libros, enlaces y herramientas relevantes:

1. Deitel, P., & Deitel, H. (2016). Como Programar en Java (10.a ed.). Pearson.
2. Villalobos, J. A. (2008). Introducción a las estructuras de datos: Aprendizaje activo basado en casos (1.a ed.). Prentice-Hall.
3. Weiss, M. A. (2010). Estructuras de datos en Java (4.a ed.). Pearson.