



**Fecha:** enero de 2026

**PROGRAMA ACADÉMICO:** INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

**SEMESTRE:** CUARTO

**ASIGNATURA:** PROGRAMACIÓN III

**CÓDIGO:** 8105429

**NÚMERO DE CRÉDITOS:** 4

### PRESENTACIÓN

Esta asignatura se orienta a la formación del estudiante para que conozca los conceptos de diseño, análisis e implementación de estructuras de datos abstractas, de los algoritmos que permiten su manipulación y de su aplicación en la solución eficiente de problemas.

Por otra parte, se orientará al estudiante a conocer conceptos básicos de concurrencia en sistemas y su integración con otros conceptos como son las comunicaciones entre dispositivos de cómputo.

Por último, se resaltarán la importancia de los diseños en sistemas con arquitectura distribuida y de los retos que presentan dichos sistemas.

### JUSTIFICACIÓN

En el mundo laboral, se requieren profesionales capaces de afrontar situaciones reales y dinámicas, que estén focalizados en la gestión y desarrollo de las mejoras continuas, por ello esta asignatura contribuye a formar las bases fundamentales, del futuro Ingeniero de Sistemas y Computación, que esté en la capacidad de aplicar los conocimientos adquiridos en la construcción de soluciones a los problemas presentados, haciendo uso de diferentes herramientas computacionales.

Luego de analizar el ámbito internacional es evidente que una de las habilidades más demandadas por las compañías de Tecnología Informática, es la de desarrollar aplicaciones, según la “Encuesta de Salarios IT 2017”, realizada por el Portal Computer World; en el campo nacional el panorama es similar teniendo en cuenta el “Estudio de salarios y profesionales del sector de Software y TI de Colombia 2016”, elaborado por MinTIC y FEDESOFTE, donde se presentan los perfiles más solicitados en la industria nacional, destacándose los relacionados con el área de desarrollo, lo que es corroborado con otros estudios que señalan el déficit de profesionales con estos perfiles, por ello la estructura de esta asignatura contribuye de manera efectiva a responder a las necesidades de esta industria.

Por otra parte, es justo mencionar que las habilidades que se desarrollan en esta asignatura, sirven como base para el abordaje de asignaturas de otras áreas que hacen parte del plan de estudios del Programa.

Por último, es importante resaltar que la lógica de programación actualmente se considera una habilidad transversal, debido a que contribuye al desarrollo del pensamiento lógico y matemático, que se requiere en cualquier campo de la Ingeniería. Existen, además, comunidades académicas que han identificado la necesidad de incluir temas de lógica de programación, desde tempranas edades en los primeros niveles de escolaridad.



### **COMPETENCIAS**

Adquirir una visión global de la Ingeniería de Sistemas y Computación en el campo laboral y construir modelos de solución algorítmicas focalizadas en las bases de la lógica y la programación, con el fin de construir soluciones reales, viables y de eficaz implementación en un lenguaje de programación.

- Conoce, diferencia y utiliza estructuras dinámicas de datos para dar solución a problemas de diferentes niveles de complejidad
- Implementa estructuras de datos en diferentes lenguajes de programación
- Entiende y aplica conceptos de programación concurrente
- Conoce los principios básicos de transmisión de datos en la arquitectura cliente-servidor
- Integra los conocimientos adquiridos en la construcción de software multiusuario
- Utiliza herramientas de software para soportar los procesos de las etapas del ciclo de vida del desarrollo de software
- Trabaja en equipo para dar solución a problemas planteados
- Genera código de calidad aplicando buenas prácticas de programación

### **RESULTADOS DE APRENDIZAJE**

- Utilizo estructuras dinámicas de datos (listas, arboles, grafos) para dar solución a problemas relacionados con el manejo de la información en diferentes contextos organizacionales, a partir del trabajo en equipo y aplicando buenas prácticas en la construcción de software.
- Aplico conceptos de programación concurrente y principios básicos de transmisión de datos en la construcción de aplicaciones multiusuario utilizando la arquitectura cliente-servidor.
- Utilizo diferentes lenguajes de programación y herramientas especializadas en cada una de las etapas del ciclo de vida de desarrollo de software.
- Leo documentos técnicos en idioma inglés y utilizo términos propios del mismo en el proceso de codificación.

### **METODOLOGÍA**

Esta asignatura será guiada en los espacios presenciales por la complementación conceptual del docente al trabajo de preparación previo que los estudiantes han realizado sobre la temática particular a tratar en la sesión; por lo tanto, un tema será abordado en cuatro momentos:

1. Preparación, consulta e investigación conceptual por cuenta del estudiante y su pequeño grupo de trabajo.
2. Tratamiento conceptual del tema en sesión del gran grupo junto con el docente.
3. Aplicación de talleres individuales y cooperativos a nivel tutorial.
4. Desarrollo de actividades de refuerzo en sesiones autónomas.

De lo anterior se verifica que en la actividad 1, el estudiante constituirá conflictos conceptuales de baja complejidad, a solucionar en el transcurso de la actividad 2, entre tanto, la actividad 4 generará conflictos cognitivos orientados a la aplicación, a subsanar con la actividad 3.

Se considera que el estudiante debe alcanzar un amplio trabajo autónomo, que posteriormente será complementado por el trabajo cooperativo de su pequeño grupo. La generación de conflictos cognitivos es importante y necesaria, para que las sesiones de gran grupo cumplan con su objetivo de afianzamiento del conocimiento.



### INVESTIGACIÓN

La asignatura por sus condiciones particulares, promueve en el estudiante la investigación formativa, de modo sea posible obtener los fundamentos con los que podrá posteriormente vincularse en el tratamiento de áreas específicas de la disciplina Informática, por medio de los grupos y semilleros de investigación dinamizados al interior de la Escuela de Ingeniería de Sistemas y Computación.

### MEDIOS AUDIOVISUALES

Salas de informática

La asignatura es eminentemente práctica, de manera que se hace necesario que todas sus clases sean cumplidas en salas de informática.

### EVALUACIÓN

#### EVALUACIÓN COLECTIVA

No aplica

#### EVALUACIÓN INDIVIDUAL

La evaluación en la asignatura está orientada a determinar el nivel de desarrollo de los procesos lógicos en el estudiante, junto con su capacidad para abstraer problemas y generar soluciones informáticas.

#### RANGOS DE VALORACIÓN (%)

El cálculo de la nota final se hará de la siguiente manera:

##### Primer corte

- 60% Evaluaciones
- 20% Talleres y exposiciones
- 20% Trabajos

##### Segundo corte

- 60% Evaluaciones
- 20% Talleres y exposiciones
- 20% Trabajos

### CONTENIDOS TEMÁTICOS CENTRALES

#### 1. ESTRUCTURAS DINAMICAS (clases genéricas)

- 1.1. Listas
- 1.2. Pilas
- 1.3. Colas
- 1.4. Utilización de iteradores
- 1.5. Análisis de complejidad

#### 2. ESTRUCTURAS RECURSIVAS BINARIAS

- 2.1. Árboles binarios
- 2.2. Árboles ordenados balanceados (AVL)

#### 3. ESTRUCTURAS RECURSIVAS N-ARIAS

- 3.1. Árboles n-arios

#### 4. GRAFOS

- 4.1. Algoritmos de recorrido (profundidad, niveles y heurístico)
- 4.2. Algoritmos de búsqueda de caminos mínimos
- 4.3. Algoritmo de Dijkstra

- 4.4. Algoritmo y camino de Euler
- 4.5. Ciclos y caminos de Hamilton
- 4.6. Árboles de recubrimiento
- 4.7. Matrices dispersas

**5. PROCESOS MULTITHILO**

- 5.1. Definiciones y conceptos básicos (Thread, Runnable)
- 5.2. Ciclo de vida de un hilo
- 5.3. Grupos de hilos
- 5.4. Estados de un hilo
- 5.5. Comunicación entre hilos

**6. TRANSMISIÓN DE DATOS**

- 6.1. Sockets
- 6.2. Envío de archivos binarios
- 6.3. Protocolos de comunicación (XML - JSON)

**7. SERVICIOS EN SISTEMAS DISTRIBUIDOS**

**LECTURAS MÍNIMAS**

Artículos y apartados bibliográficos de los diferentes temas a tratar, proporcionados por el Docente. Consulta permanente a los foros de las comunidades de desarrolladores.

**BIBLIOGRAFÍA**

- Bruce Eckel. Piensa en Java. 4ta edición. Pearson Education. 2007
- Robert Sedgewick, Kevin Wayne. Algorithms. Addison Wesley. 2011
- Robert C. Martin. Código limpio. Manual de estilo para el desarrollo ágil de software. Prentice Hall. 2009
- Martin Fowler y Kendall Scot. UML gota a gota. Publisher, Pearson Educación, 1999.
- Jorge Villalobos y Rubby Casallas. Fundamentos de programación. Aprendizaje activo basado en casos. Pearson Prentice Hall. (2007).
- Stephen Stelling y Olav Maassen. Applied Java Paterns. Publisher: Prentice Hall PTR First Edition. 2001
- Alexander Hristov. Manual de estilo de programación. Creative Commons. 2007
- Clifford A. Shafer. Data Structures & Algorithm analysis in JAVA. Pearson Education. 2012.
- Mark Allen Weiss. Estructuras de datos en JAVA. Pearson Education. 2013
- Carlo Casorzo G. Estructuras de datos abstractas en lenguaje JAVA. Universidad de Santiago de Chile.
- Joyanes A., Luis. Algoritmos y estructuras de datos: una perspectiva en C. McGraw-Hill. 2004, 655 p
- WEISS, Mark Allen. Estructuras de Datos en Java. Pearson Ed. 2000
- Lewis, J; Chase, J. Estructuras de datos con Java. Pearson. 2006
- Joyanes, Luis; Zahonero, Ignacio. Estructuras de Datos en Java. McGraw-Hill. 2008.
- LANGSAM, Yedidyah. Estructuras de datos con C y C++. Prentice-Hall, 1997
- MARTÍ, Narciso Oliet y otros. Estructuras de Datos y Métodos Algorítmicos. Pearson. 2004.
- Fowler Martin et al, Refactoring: Improving the Design of Existing Code, Addison-Wesley, 2012.
- Karavirta Ville and Shaffer Cliff CS2 Software Design & Data Structures, OpenDSA Project, 2017.
- Chawdhuri Debasish Ray, Java 9 Data Structures and Algorithms, Packt Publishing Ltd, 2017.
- McAllister William, Data Structures and Algorithms Using Java, Jones & Bartlett Publishers, 2010.
- Skiena, Steven S, The Algorithm Design Manual, Springer, 2008.
- Rusty Elliotte, Java Network Programming: Developing Networked Applications, O'Reilly Media, Inc, 2013.
- Goetz Brian et al, Java Concurrency in Practice, Addison-Wesley Professional, 2006.

En línea:

- Moreno, Juan Carlos (2015). Programación orientada a objetos. Ediciones de la U. Disponible en <https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=7927&pg=10>
- Deitel, P y Deitel, H. (2021). Java cómo programar objetos tardíos – Introducción a Jshell. (11a. ed.) Pearson Educación. Disponible en <https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=16477&pg=1>
- Joyanes, L. (2020). Fundamentos de programación. (5a. ed.) McGraw-Hill - Plus. Disponible en [https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=33138&pg=epubcfi\(/6/6\[data-uuid-c5071462f4ee4aa1bee5670c6ffa357a\]/4/2\[P%C3%A1gina+I\]/1:0\)](https://www-ebooks7-24-com.biblio.uptc.edu.co/?il=33138&pg=epubcfi(/6/6[data-uuid-c5071462f4ee4aa1bee5670c6ffa357a]/4/2[P%C3%A1gina+I]/1:0))
- Java, API especification Oracle. Disponible en <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>

Nombre del docente responsable: Jorge Gabriel Hoyos Pineda